

SyncWare News

The journal for technical
applications of T/S
computers

Volume 3 Number 6

Jul.-Aug. '86



Quantum Levels Premiers

The August/September issue of Quantum Levels is now complete. An \$18.95 yearly subscription brings you 6 issues of material exclusively for the QL. Each 16 page issue contains the same type of in-depth hardware and software information you've come to expect from Thomas B. Woods Publishing. As they say on TV, "Don't delay, order today!". Thomas B. Woods Publishing, P. O. Box 64, Jefferson, NH 03583

Timex & Other Rumors

Timex Portugal, currently having success with its Spectrum in the European market, has announced plans to bring the home computer battle back to North American shores. Rumor in the industry is that they will be marketing 2048 (Spectrum) and 1500 type machines in Mexico. They are rumored to be currently developing a QL clone.

CST and Eidersoft are also planning to put together a QL clone that they have tentatively named "Thor". The Thor will be comprised of a QL-type motherboard, but without the micro drives. In their place is to be a 3-1/2" Quad Dens disk drive. Included amongst Thor's features will be 640K of RAM, and ports for serial RS-232, Centronics parallel, a mouse, joysticks, and networking.

Amstrad Invasion

Amstrad Consumer Electronics, the British firm that bought Sir Clive Sinclair's ailing computer business, will be coming into the US market in a big way. Sears, Roebuck & Co. has announced that they will be offering Amstrad's new word processor. Billed as the affordable replacement for the typewriter, Amstrad's PCW 8256 is already a success in the European market. The unit is currently being test marketed in Los Angeles and New York. Labor Day is the tentative target date for nation wide distribution.

The PCW8256 will be marketed as a complete word processing system for small business as well as home use. It will come with a built-in floppy disc drive, keyboard, monitor, word-processing software, letter quality printer, and RS-232 port- all for \$799. It uses a CP/M operating system and has 256K of ROM. The Amstrad machine will also be upgradable to a personal computer.

All this has left our Sinclair machines out in the cold as far as Amstrad is concerned. It has dropped the ZX81 and the QL in preference to their own line of low priced computers which have been doing extremely well in Europe.

FOR YOUR SUPPORT

This column announces any software, hardware, and related modifications that are new or otherwise untested by us. If you have something of interest, we will announce it here. Send us a description of your product (keep it short). We advise readers to send a self addressed and stamped envelope, as a courtesy, to the individuals or companies for information.

Bill Bell, 596 Cherrington Road, Westerville, OH 43081, (614) 882-3883, is offering BBDOS for the Aerco/TS1000. BBDOS is a fully automatic, basic transparent, operating system that locates itself in the 8K-16K region of RAM. It is advertised as being fully compatible with all printer interfaces. \$29.95.

G. Russell Electronics, RD 1 Box 539, Centre Hall, PA 16828, (814) 364-1325 is currently mailing their July 1986 catalog. Hardware and software for the QL, 2068, and 1000 computers is available. They are offering several specials that include a free gift with each order.

"SILENT CONVERSATION: Newsletter of the Committee for the Game" publishes symbols, rule variations, and comment by players of The Glass Plate Game, a game of visually directed thought/conversation. The game began within a project of reacting photo-graphically to Herman Hesse's fictional Glass Bead Game and Castalia. A single copy of this newsletter may be obtained by sending a SASE/IRC to: Dunbar Aitkens, 1460 SW "A" Street, Corvallis, Oregon 97333.

I. B. A. Computer Users Network, P. O. Box 11174, Chattanooga, TN 37401, (615) 821-4730, is currently offering memberships to their users group. For your money you will receive a bi-monthly newsletter, access to the groups software library, and discounts on computer products sold through the group. They are also in the progress of compiling a Guidebook that will list all known sources of TS computers and producers.

Brice Road Pharmacy, 1653 Brice Rd., Reynoldsburg, OH 43068, (614) 861-3600, is handling the complete line of QL computers, peripherals, and software. They are also offering warranty and out of warranty service on the QL.

Damco Enterprises, 67 Bradley Ct., Fall River, Mass. 02720, (617) 678-2110, is marketing the Damco/Rotronics Wafadrive System for the TS2068. Featuring two 128K drives, a built in centronics I/F and an RS232 interface, plus several other extras, the system sells for \$229.95. Wafer prices range from \$2.99 for a 16K unit to \$3.99 for a 128K one.

Sharp's, Rt. 10, Box 459, Mechanicsville, VA 23111, (804) 730-9697, is offering "War in the East" and other strategic warfare games for use on the TS2068 and Spectrum computers. The games list at \$19.95 each with discounts given when two or more titles are ordered.

C. W. Associates, 419 N. Johnson Street, Ada, OH 45810, have announced that they are now Authorized QL Dealers. Their new catalog is currently available for the asking.

As I have been going through the numerous boxes of material sent me by the former editor, Tom Bent, when SWN moved its offices from Maryland to Ohio, I have managed to collect several advertisements that I assume were meant to go in this column. Because of incompleteness of the ad, the possibility of the material being out of date, or whatever, I did not include them at this time. However, the following is a list of the companies involved. If you find yourself in this list, and would like to be mentioned in a future SUPPORT column, send a current catalog or product description to the Ohio editorial address. I apologize for any inconvenience this may cause anyone.

Budget Robotics
Tucson, Arizona

Petri Electronics Enterprises
San Antonio, TX

Richard Booth
Bethlehem, PA

T. E. J. Computer Products
Los Angeles, CA

Pratt Programs
Wichita, KA

D. Lipinski Software
Roslyn, PA

J. Keene
Dallas, TX

```
*****
*
*
* *** SMART TEXT TS-2068 ***
*
* Administrative Software
*
* Data Base - Word Processing
* Mailing List - Repeat Printing
* Form Letters - Auto Letter Head
* Auto Sign Off- Labels/Envelopes
* All integrated functions.
*
* Versions For:
* AERCD DISK - OLIGER DISK
* CASSETTE - A&J Micro Drive
* Each Package includes four
* programs. The Cassette, Wafer,
* and Oli Disk versions includes
* one for the OS-64 Cart.
*
* The most comprehensive software
* for the TS-2068.
*
* $34.00 post paid- Check or MD
* Bill Jones, Gulf Micro
* 1317 Stratford Ave,
* Panama City, FL 32404
*
* Inquiries welcome
*****
```


FORUM

To start off this month's column, we'd like to submit this short program for your approval. It was submitted to us by Arthur B. Marzano. Enter it on your TS2068. It is intended for use with a color monitor although you'll get the general idea with a monochrome one.

```

10 BORDER 0: PAPER 0: OVER 1: CLS
20 FOR I=1 TO 10
30 LET A=3
40 LET X=RND*255: LET Y=0
50 IF X>=125 THEN LET A=-A
55 INK 4
60 FOR F=1 TO 50
70 PLOT X,Y
80 LET X=X+RND*A: LET Y=Y+RND*4
85 IF F>5 THEN INK 1
90 NEXT F
100 LET H=1
110 LET C=RND*40: LET D=RND*44
120 INK (RND*7)+2
130 DRAW C,D: DRAW -C,-D: DRAW
    -C,D: DRAW C,-D
140 BEEP .005,-C
150 LET C=-C: LET D=-D
160 DRAW D,C: DRAW -D,-C: DRAW -D,C: DRAW D,-C
170 BEEP .002,D+10
180 LET H=H+1
190 IF H=20 THEN GO TO 210
200 GO TO 110
210 NEXT I
220 FOR I=1 TO 7: PRINT AT 19,0: INK F: OVER 0:
    "HAPPY 4TH OF JULY FROM SWN!":
    PAUSE 50: NEXT F
230 PAUSE 0
240 GO TO 10
    
```

New SyncWare Kid

Anyone who, as a child, has moved from one neighborhood to another, or had someone new move into their neighborhood, will recognize what I'm going to refer to as the "New-Kid-on-the-Block" Syndrome. It's always a time of testing and proving for all involved. Your new found peers want to know who you are, where you're from, and what kind of toys you have. Plus there's often a few fights just to establish who's going to be "top dog". Hopefully we won't have any fights, but I believe our readers are entitled to some answers about who this "new kid" in the SWN Editor's chair is.

My name is Jeff Moore. I reside with my wife and two sons at the same address as our new Editorial offices: 602 S. Mill St., Louisville, Ohio 44641. I hold a degree in Electronics Engineering Technology from the Ohio Institute of Technology, Columbus, Ohio. I'm employed by a local steel products processing plant. Because I work a "swing" shift at the factory, my office telephone hours will be from 9 AM to 11:30 AM and/or 8 PM to 10:30 PM, depending on the day of the week and/or the week you may call. The balance of the time is for family, sleeping, and eating. There is a telephone answering machine, so please feel free to leave a message. My telephone number is (216) 875-1257.

My "toys" are, of course, my computers. Presently we have the following set up:

- 1) a 2068 w/Aerco FD-68, dual double-sided disc drives, and a Tasman type "B" CIF
- 2) a ZX81 w/Aerco FD-ZX, dual single-sided disc drives, Memotech CIF, and a Byte Back 64K RAM
- 3) a TS1000 w/Compusa FD-102 (the Compusa board is in need of repair- anyone want to help?), Oliger Video, and Oliger 64K RAM
- 4) a "bare-bones" TS1500 w/2040 printer
- 5) I have access to a TS1000 w/Aerco FD-ZX and CIF
- 6) and when my spouse allows me to use it- a QL

Along with the above, mostly due to the generosity of our readers and advertisers, we have a good collection of hardware add-ons that was turned over to me by the former Editor, Tom Bent. We can handle about any software that you care to send.

The past few weeks have been used setting up the new Editorial Office and going through all the material Mr. Bent had accumulated in his two years as editor. According to my calculations, I now have 6,573 more cubic feet of material and hardware stuffed into my office than I have cubic feet of office. Which partially explains the tardiness of this issue. Even so it looks as though the "new kid" is finally organized. Although I'm the "new-kid-on-the-block", I've already set some goals. They are to give you the best magazine on the market for Sinclair/Time computers. All the time. And as the other guy advertises- ON TIME.

SUM, Time Designs Merge

As in the case of Mark Twain when he said, "The rumors of my death have been greatly exaggerated", the rumors of the demise of SUM magazine have also been overstated.

Joe Williamson, Editor of SUM, and Tim Woods, Editor of Time Designs, announced the merger of the two Timex/Sinclair orientated magazines June 30th. SUM will publish its July issue before the merger is finalized, according to Editor Williamson. As a result of the merger, it is anticipated that Time Designs will increase its size by about four pages. Williamson has been added to the Time Designs staff and will be handling hardware projects and reviews.

Time Designs will fulfill any unexpired SUM subscriptions with its magazine. In the case of an individual having a current subscription to both SUM and Time Designs, their Time Designs subscription will be extended by the number of issues remaining on the SUM subscription.

Q & A

Dear Editor,

Enclosed is my renewal for two years. I enjoy SyncWare News even though some of it is over my head. I started out with a ZX-81 four years ago. I bought a TS 2068 two and a half years ago. I have used the 2068 until the "Caps Shift" keys are just about unreadable. Please remember that all of your subscribers are not computer wizards but we do enjoy working with them.

I have a Tasman I/F and use an Epson LX-80 printer. Before I found David Ridge's TASMAN+QUADRA CHART fix in SWN 2:6 I could not figure out how to dump the

screen to the printer. I use CURVE TRACING-2068 from SWN 3:2 for simple wiring diagrams. I am now trying to figure out how to add letters and numbers to the TRACING. To me the above two programs payed for my subscription to SyncWare News. Keep up the good work. Just remember those of us that have picked up on computing by sitting down at the keyboard and enjoying the art as we learn.

Les McCullough
Amarillo, TX

Dear Les,

Thank you for your encouraging letter. I'm glad you enjoy our magazine. The fact that some of it is "over your head" is good to hear in a way. That means we're presenting challenging material with enough "meat" to keep it from being boring. However, we're also aware that not all of our readers are "computer wizards". In the future we're going to try to include more articles geared to those just getting started with the technical aspects of the TS computers or are not heavily involved in hardware hacking. Our goal has been and will continue to be bringing you the best TS technical magazine on the market.--jm

Ddear Editor,

A few days ago I bought a color TV set with a composite video input on it. Now when I use it with my TS 2068 (computer monitor output to composite input on TV, TV in video mode), there is distortion taking place on the screen when I use some BORDER or PAPER colors. White PAPER with a black or blue BORDER results in the vertical edge of the PAPER to be wavy or distorted. The same applies when a white BORDER is used with black or blue PAPER but is not as severe. Let me know if you have any suggestions to correct these problems.

Another problem occurs if I want to PRINT say a bunch of squares in blue INK on a yellow PAPER. The squares appear to be black. There is also a "shadow" phenomena, ie. if I PRINT squares in green INK on yellow PAPER. A yellow shadow will appear at the square's left edge and a brown shadow will appear at it's right edge.

Finally, when in the "normal" mode (white PAPER and black INK), there is always some annoying "rainbow" colors following the letters on the screen.

The TV itself is of the Electronic Tuning System type, the kind with push button tuning. There is a fine tuning control for each channel and automatic fine tuning once the control box door is closed. Am I correct in assuming these are not operating in the video mode? The rest of the controls are those you would normally find on any TV receiver.

I hope you have enough information to visualize my problem.

Jean-Claude Touzin
P. Q. Canada

Dear Mr. Touzin,

Your problems with the 2068 monitor display are quite common. Unfortunately, there's no one thing that can be done to give you a crystal-clear display, short of buying a RGB interface and RGB monitor.

First the matter of the distortions with a black border. This is a common problem with many monitors, especially those designed primarily for TV use with a monitor option "tacked on". It occurs because the BLACK level (no light) voltage of the videosignal is quite close to the sync-pulse level; some monitors get "confused" by this. Sometimes this can be eliminated or reduced by the careful adjustment of the contrast and the horizontal hold controls. If not, your best bet would be to stick with colored backgrounds (e.g. blue). It has been reported that this can be eliminated on some monitors by direct-coupling the video output. On the 2068, this can be done by shorting C58 (inside the modulator case). I should mention, though, that in the event of a monitor malfunction, this could damage the video driver transistor Q2, and conceivably other parts of the computer.

The second problem, shadowing on the edges of certain colors, appears to be a characteristic of the 2068 and other computers that use the LM1889N modulator chip. Again, it seems to be worse with monitors that double as TVs, so short-cuts in

The SyncWare News

Thomas B. Woods
Jeffrey D. Moore
Thomas J. Bent
Fred N. Nachbaur

Publisher
Editor
Associate Editor
Technical Director

All subscription information, billing and ad artwork should be addressed to:

SyncWare News
PO Box 64
Jefferson, NH 03583

Phone: (603) 586-7734

Article submissions, Forum, Support listings, Classified ads and any questions you may have concerning the use or everyday abuse of Timex Sinclair computers may be sent to any of the editorial offices.

Jeff Moore
602 S. Mill St.
Louisville, OH 44641
(216) 875-1257

Tom Bent
9016 Flicker Pl.
Columbia, MD 21045
(301) 730-7187

In Canada

Fred Nachbaur
Mountain Station Group Box C-12
Nelson, BC V1L 5P1 Canada
Phone: (604) 354-3858

Back issues of SWN are available for \$3.50 each.
Volume #1 (for the TS1000) is available for \$16.95

SWN is done entirely on Timex Sinclair computers. Submissions are preferred as word processed text files (so we need not reinvent your article) on good tape. Memotext for the 1000, Mscript for the 2068, QLWP for the QL are the preferred word processors. We can also handle RCPM files done in WordStar (or Ascii format CPM files). Tasword files must be done in a 52 column format. Programs must be submitted on tape, so that we can verify that they will run. Label the tape itself, in case the documentation gets separated from the tape. If you do not have a word processor and have a good idea, then don't hesitate to write up your thoughts and submit them. SWN pays about \$40 per page for original articles. This amount varies slightly from issue to issue and is paid when the article is published. Write for more information.

monitor design could have something to do with this also. Sometimes the "sharpness" control can be used to make this less obvious. You might also be able to reduce this effect by adjusting the chroma crystal trimmer C18, located inside the little modulator box at the upper left corner of the 2068 board. Use a non-metallic screwdriver (available at electronics parts distributors) to slowly turn the little slotted adjustment. Note its original position so you can return to it if no improvement occurs. This might also help bring in your blues on a yellow background.

The "rainbow" colors with white characters on black is another problem relating to the monitor. You might have noticed the same effect on television, such as on pin-stripe suits (which is why such costuming is usually avoided in made-for-TV productions). The more expensive monitors have a "comb filter" which helps to reduce this. A competent TV repair person might be able to reduce this effect. Your best bet, though, with white/black screens is to simply turn the color control all the way to shut off the colors entirely. Or, again, go to a RGB system, in which each color is sent to the monitor on a separate wire.

Lastly, "automatic fine tuning", or in fact any sort of tuning, has nothing to do with monitor operation. The whole idea of a video monitor is to completely bypass the entire tuner section. It's like your stereo; when you're playing a record, the radio's tuning dial has no effect. Just as the phonograph is connected directly to the audio amplifier, so also is the "monitor" video signal connected directly to the video amplifier.

To sum it all up, I'm afraid there's not too much you can do. These and other display deficiencies are on reason I prefer to use the ZX81/TS1000 with the Oliger video upgrade and a monochrome monitor. It seems that what looks fine to some people is completely objectionable to others. Don't be afraid to use the controls provided on your set to get the best picture you can with whatever program you're running at the time. Adjust your system as best as you can, and "grin and bear it".-fn

Dear Editor,

What is confusing (to me) is probably very simple for you. When you write ZX 81 do you invariably refer to the TS1500? Specifically, does the Oliger TI Video Upgrade refer to the TS1500?

Sincerely,
James Smith
Bath, Maine

Dear Sir:

Your question about TS1500/ZX81 compatibility is an excellent one. In general, any program that was written for the ZX81 (aka TS1000) will run without change on the TS1500. There is only the rare exception; one of the most well known is the "Z-XLR8" Fast-load tape package. Until recently, it was thought that Z-XLR8 would not run on the TS1500. I have since discovered that this is only because of a minor hardware difference. By using a "reversed" load cable (i.e. connected "backwards") Z-XLR8 works fine on the TS1500 (see figure 1).

However there are capabilities built into the TS1500 that are absent on the ZX81. Specifically, the

TS1500 is capable of high-resolution displays far surpassing similar packages for the ZX81/TS1000. Greg Harder's article in SyncWare News #3:4 describes this. His "Hi-Res Extended Basic" (available from Silicon Mtn. Computers, C-12 Mtn. Station Group box, Nelson, BC V1L 5P1, Canada) carries this technique into an actual programmer's utility. It should be noted that these programs WILL NOT run on the ZX81/TS1000, whereas hi-res routines written for the ZX81/TS1000 will run on the TS1500.

The only other consideration is memory. Most ZX81/TS1000 owners have long since upgraded to 64K, so the occasional program (such as Memotext V3) presumes a 64K machine. 64K packs, as well as most other hardware add-ons, for the ZX81/TS1000 will work with the TS1500, but some of them are too long and interfere with the connectors on the back of the TS1500. On the '81/1000, as you may know, has the connectors mounted on the side. So before you buy hardware add-ons for the TS1500, you should be sure that the mechanical dimensions are compatible with the 1500.-fn

FIGURE 1: Z-XLR8/TS1500 ADAPTER

Cut a standard cable in the middle, then splice it together "backwards", i.e., the shield of one end goes to the center conductor of the other, and vice versa. Another way is to make up a backwards cable from scratch.

ONE WARNING:

Be sure to disconnect the MIC cable when using the backwards cable! This is ALWAYS a good practice, but it's an absolute necessity when using an inverted load cable. Otherwise you will short the EAR output; at best, it won't work, and at worst it might damage your recorder.



Dear Editor,

I have a problem that your readers may be able to solve.

Does anyone have a modification to the AERCO interface software for the TS 2068 to operate the new AK Royal Lettermaster daisy wheel printer?

I recently purchased this printer. It is very reasonably priced and should have wide distribution by now. AERCO has not responded to my request for help.

I am using an Olivetti inkjet printer with this combination (AERCO and TS2068) with full performance. I'm a long time subscriber and appreciate every word of your publication.

Charles Grosh
Pacific Palisades, CA

Dear Charles,

Thank you for your letter. I hope SyncWare News and you continue together for a long time. I was unable to find an answer to your printer problem but I'll put the question to our readership. Does anyone have an answer to the AERCO/Royal printer problem. If you do, let us know and we'll pass the word along.-jm

Knighted SAVES

Dear Editor,

I had been having trouble for some time while trying to save programs on my TS 2068 computer. I had tried implementing several possible solutions without any success. In a catalog I received from Knighted Computers of Fulton, NY was a suggestion that finally solved my problem. I'd like to share it with your readers.

The video generator in the 2068 is located near the left side of the computer. If your cassette recorder is also positioned on the left side of the computer, when doing a save, you may get interference that may cause an error when trying to load a tape. Knighted went on to suggest that the cassette be always placed on the right side of the computer, and as far away from the TV/monitor and computer as your cables will allow. This applies not only to cassette recorders, but to any data storage device you may be using.

I never had any trouble loading. Following their advice cured my saving problems that had existed for over a year.

Sincerely,

E. Swallow
Manchester, CT

More On Spikes

The purpose of this letter is to comment on and expand upon the article on page 3 of SWN 3:4 concerning voltage spikes.

I have Relax Technology's power control 3 unit under my monitor on the TS-2068 and everything (CPU, monitor, disk drives) is consequently "protected" from most electrical problems. Since the spike protector is only one power cord it also makes it easy to unplug everything if a severe electrical storm is in the area. As I said before, "everything" is protected EXCEPT my telephone line coming into the modem. I knew I was going naked on the modem but really didn't give proper consideration to the phone line until a really great electrical storm passed over the neighborhood at 3 AM on Saturday night. First thing Sunday, I turned on all the computers in the house to be sure they were working and breathed a sigh of relief. I tried for two days to reach a local bulletin board where I knew I had a message waiting and got only busy or no answer messages before it dawned on me that something was wrong. Sure enough the static coming in the phone lines had rendered the modem deaf and dumb. I am thankful that the MOV just inside the line connection stopped it there and it didn't go into the edge connector and on to the CPU or some other disaster area. I now have a Tripp-Lite modem spike arrestor, a new TS-2050 modem on order, and, thankfully a promise of a check from my homeowners insurance carrier for the replacement modem, plus a new respect for the importance of surge protection on ALL of my computer equipment.

Your readers might like to know of the importance of telephone line suppressors as a addenda to your former article. Reading about it is an easier way to learn than I had.

Sincerely,
Phillip Stevens
Des Moines, Iowa

Print Message Notes

The article on using the message routine in the 2068 that appeared in the March-April issue was useful. However, two problems exist with the routine.

First, if a print statement is before the message routine call (same line) the 2068 prints your message on the upper screen (the exact location depends on the last print location used). This can be corrected by calling the 2068 routine referred to as SELECT with A=0, setting the channel to the lower screen.

The second problem is the position of the flashing cursor at the end of the message. This, along with the echo of any keys pressed immediately after the message, tends to be unprofessional in appearance. To eliminate this, reset bit 3 of FLGS (prepares the computer to receive a keyword) and set bit 5 of TVFLGS (makes cursor non-visible and clears the lower screen when key pressed and echos it).

The following code is the corrected version. Note that register A must contain the message number when this routine is called.

```
ERROR PUSH AF      F5      : save message no.
LD A,00            3E00     : set channel for
CALL 1230          CD3012   : bottom of screen
POP AF            F1      : recover message no.
LD DE, MSSG       11 MSSG   : MSSG=start of messages
CALL msg          CD3F07   : output message
LD HL, flgs       213B5C   : reset bit 3 of flags
RES 3, (HL)        CB9E    : (for keyword)
INC HL            23      : set bit 5 of tvflags
SET 5, (HL)        CBEE    : (for clear bottom)
JP 0E2F           C32F0E   : jump to control loop
MSSG              Code of message(s) starts here
```

Sincerely,
James Hunkins
Rapid City, SD

Canada Customs

If you're supplying computer goodies to customers or pals in Canada, the following tips will help the fellow on the receiving end cope with the Customs service.

- 1: Go to the trouble of filling out a Customs Declaration. It will take you less than a minute, and could save your customer lots of time and hassles.
- 2: If you're sending hobbyist software or hardware (as most if not all of the stuff for our machines should be considered), MARK IT AS SUCH. When customs people see the word "Computer," they immediately think IBM stuff worth hundreds of dollars. Indicate that the stuff you're sending is for experimenter use; the watchword is HOBBY.
- 3: When sending hardware or parts, mark it as COMPUTER hardware or parts. Though it doesn't make any sense, the duty rate for ELECTRONIC parts is over 3 times as high as the rate for COMPUTER parts, even though most components could logically be classified either way.
- 4: Mark packages with programs on EPROM as COMPUTER SOFTWARE (which it is), not as computer hardware or parts. This is because hardware is dutiable, software is not.

Author Missing

While I was going through the many boxes given me by the former editor, Tom Bent, I came across a good article on Machine Code Implementation of Read, Restore, and Bump for the T/S1000. The only problem is I have no idea who the author is. If the author is out there and would like to be published, drop me a line.

Tom's Soapbox

SINK OR SWIM, WE'RE ALL IN THIS TOGETHER

One of our old friends related a story I'd like to share with you because it's important and because it started me thinking. I am not going to mention any names, partly to help our friend avoid an embarrassing situation and partly because we probably ALL have done this at one time or another. And when I say ALL, I mean me, you, our competitors, our friends, our associates, everybody. As you read this, just put yourself in this situation, and then THINK.

"...I attended a meeting of the ---- T/S User Group two weeks ago. Drove down with ----. Counting him and me, there were only four of us present. We spent the entire three hours copying programs from the club library. So much for the activities and morals of the local chapter. Brought with me a couple samples of ----, hoping to garner a few subscriptions, and danged if one of the ingrates didn't take one of 'em upstairs to the 10 cents a copy machine and run off what he wanted out of it. The group leader bemoaned the fact that they did not have an editor for their newsletter. I asked, "what newsletter?" and he said that he had been issuing a single page letter

advising everyone of the meeting date and urging them to attend. Claimed if they only had an interesting regular type newsletter that more members would attend the meetings."

We as Timex hobbyists are an endangered species. When they quit making our computers, we lost a lot more than just a great computer. We also lost the influx of new members, new talent, and new ideas to our ranks. And to make matters worse, the human talent that we do have is bound to grow only smaller in time. We lose interest because of lack of stimulation, we cry because there's no support, yet we refuse to take the bull by the horns and do it ourselves. We copy books, magazine articles, programs, and even hardware. Sure we save money. How could it hurt the author that much by copying just this one time? Besides, everybody else does it. Why not me? We are ALL guilty of copying. We have grown so accustomed to it that we don't even think about it. It just happens.

I have received programs worth hundreds of dollars. Sometimes I was grateful for the "gift". Can anyone tell me how or to whom gratitude should be expressed for such a gift? Other times I wondered how I would feel if I were the author of the program I had received. I can tell you as a software producer that everytime I learn about someone who has copied one of my programs, it hurts. I am sure that everyone else who has been in the same situation will agree. It doesn't matter what is copied, or why, or who the person is who loses a small chunk of income as a result. In the end we ALL lose because every time a product is pirated, the victim of the piracy takes one step closer to giving up and moving on. It's not a question of legality or of money. It's a question of morals and survival. Where would we be without the

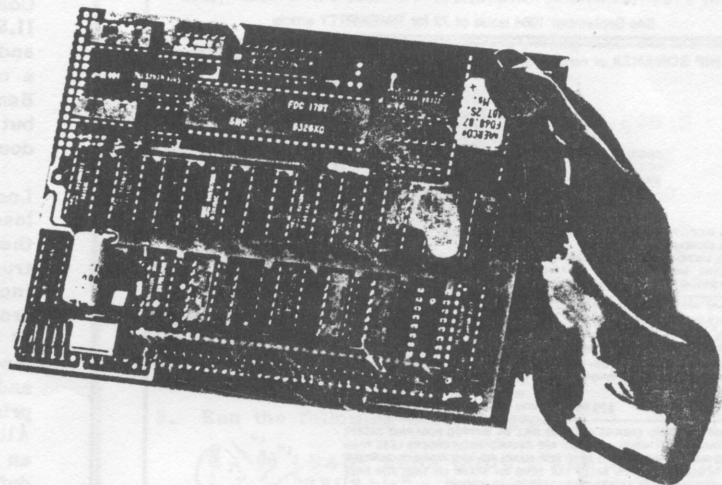
THE AERCO FLOPPY DISK INTERFACE RUNS 3.2 MEGABYTES OF ON-LINE CP/M SOFTWARE... 50% FASTER THAN KAYPRO.

Don't forget our printer and serial interfaces too!

AERCO

ACME ELECTRIC ROBOT CO.

BOX 18093 AUSTIN
TEXAS 78760-8093
(512) 451-5874



CP/M is a registered trademark of Digital Research, Incorporated.

E. Arthur Browns, the G. Russells, the Time Designs, the Byte Backs, the Aerocos, the Oligers, the Kingsleys, the Currys, or Sunsets, or Knightseds, the Zebras, or any one of the dozens of other small-timers who for some insane reason cling to our beloved little machines?

I know that YOU are interested in preserving our ZX/TS hobby for as long as possible. If you were not, you wouldn't be reading this. You know that I share your interest, or I wouldn't be writing this. We are different from the rest of the computer world. In the land of IBM it's mass marketing all the way. There's hype and fast talking. There's always someone new coming along. There's new programs so expensive you can't afford to buy them. There's cut-throats who'll stab you in the back the first chance they get. In the land of Timex, we're all in this together. We can sink or we can swim. The choice is ours. Think about how good we have it. Would any one of us want to lose it?

Circuit Board Hint

After building circuit boards from artwork or kits, it's a good idea to clean all the rosin from the board. This makes it easier to see bad connections and shorts, and also gives a cleaner appearance. You may have been using acetone (acetyl ketone or propanone); while it does work very well, it has some definite drawbacks. It is highly flammable, its vapors are smelly (not to mention toxic), and it attacks many plastics (especially styrene and blends like ABS which contain it).

A much safer substance is methanol, also called "methyl alcohol," "wood alcohol," or "methyl hydrate." It is not nearly as flammable, much less

toxic, virtually odorless, and works almost as well in removing rosin. Furthermore, it is harmless to plastics, so you can use it as a "bath" for your newly-completed boards without worrying about components literally dissolving away. Simply pour some into a dish containing your board, and cover it with a bowl; a few minutes later, remove the board, dry it, and inspect it for questionable connections.

Methanol is the same stuff that is mixed with gasoline in making "gasohol," so if you have a purveyor of this substance in your area you can probably get it very cheap (under \$1 a gallon). If not, you can get it at auto-supply stores under the name of "fuel-line de-icer," albeit at a much higher price.

An even better substance (non-flammable and less volatile) is 1,1,1-trichloroethane, also called "methyl ether" or "safety solvent." It has a stronger scent than methanol, but is not nearly as unpleasant as acetone. It is available from most industrial chemical suppliers at lower cost than acetone. Freon TF is also good, but is expensive and a little too volatile to be usable in a "bath."

-fn

WORD SINC II.5 & WORDFONT TS1000 Review

Distributed by:
Peter McMullin
2340 Queen St. E.
Toronto, ONT.
Canada M4E 1G9

Russell Electronics
RD 1 Box 539
Centre Hall, PA 16828
(814) 364-1325

If you're looking for a general use word processor program for home use, WORD SINC II.5 might just be for you. Although several people who have used the earlier versions of WORD SINC, including myself, did not care for the program, this adaptation has several features that make it worth buying.

Comparing WS II.4 and previous versions with the WS II.5 word processing system (consisting of WS II.5 and another program called WORD FONT) is like making a comparison between a Ford Pinto and a Mercedes-Benz. Both word processors will get the job done, but WS II.5 allows you to compose and print out documents with ease and style!

Looking over the User Manual, a 24 page volume loaded with information, one begins to appreciate the amount of effort that has gone into producing a truly "user-friendly" word processor environment. Included, along with the general description of the program and how it works, are detailed descriptions of how to use the commands, directions to make modifications to allow use of fast loader programs and how to easily install software patches for other printer I/Fs, and an assortment of useful charts. All this is accompanied by examples and written in an easy to read and understand style. This manual is definitely one of the best I've seen.

The first half of what I've referred to as the WS II.5 Word Processing System is the WORD SINC II.5 program itself. WS II.5 is a powerful, although easy

WOW! AT LAST!! A VERY AFFORDABLE COMPUTER
AT A VERY AFFORDABLE PRICE

POWERFUL FULLY PROGRAMMABLE WITH 2K OF MEMORY—PORTABLE—8-7/8" x 1-3/8" INCH MODULE SINGLE-KEY ENTRY COMMANDS—DURABLE 40 KEY MEMBRANE TYPE KEYBOARD—Z80A BASED FOUR CHIP DESIGN—EDUCATIONAL—UNIQUE SYNTAX CHECK REPORT CODES FOR ERROR IDENTITY—GRAPH DRAWING AND ANIMATED DISPLAY—ACCURATE TO 9-1/2 DECIMAL PLACES FOR FULL RANGE MATH AND SCIENTIFIC FUNCTIONS—AT AN AFFORDABLE PRICE

WE CANNOT TELL YOU THE MAKE OF THE COMPUTER BUT IT WAS MADE BY A FAMOUS WATCH COMPANY THEY USED TO SELL FOR \$99.95

WE BOUGHT OUT WHAT THE FACTORY HAD LEFT IN STOCK AND HAD TO REMOVE THE LABELS THESE UNITS ARE UNPACKAGED LESS THE 9V WALL ADAPTER AND MANUAL BECAUSE THIS IS A DISCONTINUED ITEM THERE IS NO WARRANTY

GET THEM WHILE THEY LAST LIMITED SUPPLY

BUY 1st UNIT FOR \$19.95 BUY 2nd FOR \$16.95 9V DC WALL ADAPTOR \$4.95
BUY THE 3rd UNIT (NON OPERATING PARTS) \$10.95 MANUAL (OVER 150 PAGES) \$2.95

See September 1984 issue of 73 for TIMEX/RTTY article

CHIP BONANZA (AT THESE PRICES THEY ARE A STEAL)

2708	\$1.00 EA OR 10 FOR \$ 9.00
2718	\$2.25 EA OR 10 FOR \$20.00
2732	\$3.25 EA OR 10 FOR \$32.00
2794	\$4.00 EA OR 10 FOR \$35.00
27138	\$7.00 EA OR 10 FOR \$60.00
8002	\$4.95 EA OR 10 FOR \$45.00
8810	\$1.95 EA OR 10 FOR \$18.00
88A09	\$5.95 EA OR 10 FOR \$50.00
88A21	\$2.95 EA OR 10 FOR \$25.00
4116	(or equivalent) 8 FOR \$ 5.98
4134	9 FOR \$12.95
XR 2211	(SPECIAL) \$ 2.95
TMS 9900NL MICRO-P 84 PIN 8 BIT DIB + 18 BIT CPU	\$ 4.95
TMS 9901NL MICRO-P PSU	\$ 2.95
TMS 9904NL MICRO-P CLOCK GEN AND DRIVER	\$ 5.95
TMS 9918NL MICRO-P COLOR GRAPHICS AND DISPLAY	\$ 9.95
KEYBOARD (994) 48 KEYS MEASURE 4 x 10 (HIT-TEK)	\$ 9.95

HAL'S COMPUTER GOODIES

Times printer (reg. \$99.95 now \$59.95)

Coleco cassette player (battery op) \$29.95

9V @ 850 Ma adapter (needed for 18K RAM pack) \$7.95

18K RAM pack module, new \$29.95

18K RAM pack module, refurbished \$19.95

RCA TV interface cable \$1.25

Dual jack cassette interface \$2.50

TV/Computer game switch \$3.95

Cassette Software:

I have cassette software send for Hal (reg. \$9.95 \$14.95)

Hal's Special Price: 1 for \$3.95, 3 for \$10.95, 8 for \$19.50, 10 for \$29.50, 20 for \$50.00

or let Hal select 25 different cassettes (my choice) at 25 for \$50.00!!!

APPLE II and APPLE II+ COMPUTER MAINFRAMES (fully populated): \$150

Power supply, case and keyboard, separately available Call or Write

Unit as described above, fully assembled & tested \$350 plus shipping

APPLE POWER SUPPLIES \$29.95

SHIPPING INFORMATION: ORDERS OVER \$25 WILL BE SHIPPED POST-PAID EXCEPT ON ITEMS WHERE ADDITIONAL CHARGES ARE REQUESTED ON ORDERS LESS THAN \$25 PLEASE INCLUDE ADDITIONAL \$2.50 FOR HANDLING AND MAILING CHARGES MICHIGAN RESIDENTS ADD 4% SALES TAX SEND 20¢ STAMP OR SASE FOR FREE FLYER CANADIAN ORDERS ADD \$5.00 POSTAGE IN US FUNDS

HAL-TRONIX, INC.
P.O. BOX 1101 - DEPT. H
SOUTHGATE, MICH. 48195
PHONE (313) 285-1782

"HAL" HAROLD C. NOWLAND
WBZXH

to learn and use, text and graphics processing program. It's features include text entry, full screen edit, text listing, searching, block move and delete, and two string replacement commands. The print command allows you to preset both right and left margins (left margin set being a feature missing in WORD SINC II.4) and tab positions, enter the desired number of lines per page, and the number of characters allowed per line. Also included in the program is a screen dump routine that takes an exact image of the video screen and prints a hard copy of it with a suitable HRG (High Resolution Graphics) full size printer.

The second "half" of the system is a program/data base called WORD FONT. This program is loaded and run before WORD SINC. WORD FONT allows you to store the bit images of alternate character sets in memory to be used during the print routine of WORD SINC to allow different letter styles or fonts to be produced by the HRG printer. Included are fonts for computer type printing like on the bottom of a check, a script font, two serif types, and two sans

serif types. You may also create your own font, or alphabet. (You D and D fans could even write your clues, ect. in Orcish). All fonts can then be printed normal, double width, inverse, or even double width inverse.

There were, in my opinion, three major short-comings present in WORD SINC II.4. They were lack of a left margin set, no word wrapping (a routine that prevents a word from being split at the end of a video line), and sometimes being able to out type the program. WORD SINC II.5 has the margin feature, I can still at times out type it which leads to crashes, and it doesn't have word wrap. If your in a business application where you need a "mail-merge" or data substitutions, Fred Nachbaur's MEMOTEXT is better. However, the WORD FONT feature in the WORD SINC II.5 SYSTEM, can bring new life and enjoyment to producing "one-shot" letters, posters, ect. I feel that for that purpose the advantages of the program far out weight any short comings it has.

Reviewed by Jeff Moore

PUT THOSE EXTRA BANKS TO WORK

Charles Stelding
Tyler, Texas

Several recent developments in the Timex 2068 hardware have made our computers much more flexible and powerful. Extra memory banks available to us now give us room to do things never before possible. Such extra memory banks are available on the AERCO disk interface and RAM cartridges.

But how can we use this extra memory? What practical use is all that room that is now available? As a start, I have a few ideas that work and I have enjoyed experimenting with. What follows in this article is one use for that extra memory.

The ZEUS assembler (reviewed in SyncWare News 3/3) uses memory from 57344 to 65279 plus a source code which can be displaced to a different location but normally starts at 32768. Now, what if you want to place your written code at those locations, especially in the 57344+ area? If you write your code as ORG 57344, the program will crash because you are writing over the ZEUS code.

By putting the ZEUS program in another bank, you will have all of home bank available to put your code (anywhere in home bank above the Basic area).

Here are the steps to accomplish this:

1. Load ZEUS into home bank.
2. Enter the following program using your ZEUS assembler (adapted from David Hill's FD-68 USER, Vol. 1,1):

```
00010 ORG 32736
00020 XFER LD HL,#FFFF
00030 LD BC,#8000
00040 DI
00050 LOOP LD A,(BC)
00060 PUSH AF
00070 LD A,#F0
00080 OUT (#F4),A
00090 POP AF
00100 LD (BC),A
00110 LD A,1
```

```
00120 OUT (#F4),A
00130 INC BC
00140 PUSH HL
00150 SBC HL,BC
00160 POP HL
00170 JR NZ,LOOP
00180 EI
00190 LD A,#F0
00200 OUT (#F4),A
00210 RET
```

3. Save this XFER code starting at 32736 with 32 bytes (i.e. AERCO disk: MOVE "XFER.BIN",32736,32).

This routine will put any code beginning at 32768 to the end of home bank into the dock bank with chunks 4-7 enabled (32768 to end). To transfer the code back from dock to home bank, you must load different values to A in lines 70, 110 and 190 (see step 7).

4. Change the ZEUS code by RUNNING the following program:

```
10 DATA 0,205,206,255
20 DATA 213,197,88,6,255,205,5
9.99,193,209,42,15,253,201
30 FOR J=60515 TO 60518: READ
a: POKE J,a: NEXT J
40 FOR J=65486 TO 65499: READ
a: POKE J,a: NEXT J
50 MOVE "ZEUS rev.bin",57344,8
157: STOP
```

These changes allow ZEUS to write code to home bank while working in the dock bank. Now save the revised ZEUS starting at 57344 with 8157 bytes (i.e. AERCO disk: MOVE "ZEUS.BIN",57344,8157).

5. Run the following basic program:

```
60 LOAD "ZEUS"CODE : REM AERCO disk= CAT
"ZEUS.bin";: Note: this is the revised
code in step 4
70 LOAD "XFER"CODE: REM AERCO disk= CAT
"XFER.bin",
```

```

80 RANDOMIZE USER 32736: REM transfer ZEUS
   into dock
90 RANDOMIZE USER 57344: REM initiate ZEUS
95 STOP

```

6. Now that ZEUS is up and running, write any routine with an ORG say, for example, at 57344 which is where ZEUS code happens to be in the dock bank. Press A to write your code beginning at ORG and Q to quit ZEUS. Enter OUT 244,1 which enables home bank. PRINT PEEK at 57344+ to observe that your finished code is in home bank. You have written your code where ZEUS would have been in home bank which is the goal of this particular use of your dock bank. You may now save the code by using the normal save.

7. If you want to save your source code which is normally at 32768+, you must put it back into home bank. You do this when ZEUS is running by pressing Q and RUN the following Basic lines:

```

100 POKE 32748,1: POKE 32752,24
0: POKE 32764,1
110 RANDOMIZE USR 32736

```

The source code can now be saved starting at 32768 with the proper number of bytes. If you happen to be

in home bank (by entering OUT 244,1 at step 6) and want to enable the dock bank again to get back to ZEUS or your source code, then OUT 244,240. This will enable the dock bank, chunks 4-7.

Another use of the dock bank which I have working is to put the Spectrum code at chunks 0 and 1 and run Spectrum programs using the AERCO disk interface with its dock bank. No changes in hardware or switches are necessary!

You can also change the character set bit map in the 2068 by putting the ROM into chunk 1 of the dock bank. You must transfer the AERCO disk code located in chunk 1 to another location such as to chunk 7 and the put it back before saving the program to disk. Of course you can put the bit map elsewhere in memory, but it takes up 768 bytes of RAM which might be in the way of a large program like PRO/FILE or VU CALC. I use a fat script routine to change the characters into more readable script for those programs.

I would be interested in other uses for our extra banks or if you want to drop me a line, I'll be glad to hear from you.

BASIL'S COMPENDIUM

Basil Wentworth
Bloomington, IN

Let A = ...

This installment is going to show you how to assign values to a number of variables--or, as computer jargon has it, to load registers. But first, our little digression.

Here is a way you can build up a huge block of nulls--thousands of them, if you wish (and if your RAM is big enough). We'll find a practical use for them in our next issue, when we learn how to use machine language to store a screenful of material in a single REM (or PRINT, or LPRINT) statement.

But first, one caution. After you create the MEGA-REM, you must never DELETE the line immediately following it. You may EDIT that line however much you wish, even to the point of reducing it to a naked 2 REM. But if you try to DELETE it, you're in grave danger of a crash.

With that one caution in mind, here's what to do. First, enter the machine code routine shown in Figure 8-1, and then build up a number of sub-assemblies that will be joined to make the MEGA-REM. These sub-assemblies will look like this:

```
50 REM.....
```

Once you have one of them entered, you can easily make lines 49, 48, etc. identical to this line, just by bringing it down into EDIT position and changing the line number. And then establish an end point for the series by

```
51 REM "
```

There are a couple of restrictions on the subassembly lines:

```

16514 195 ? JP
16515 150 16534
16516 64 AND
16517 62 Y LD A,
16518 11 " 11
16519 33 S LD HL,
16520 191 16575
16521 64 AND
16522 17 LD DE,
16523 180 16564
16524 64 AND
16525 1 LD BC,
16526 0 0
16527 0
16528 35 7 INC HL
16529 3 INC BC
16530 190 CP (HL)
16531 32 4 JR NZ,
16532 251 CLS -5
16533 201 TAN RET
16534 205 LN CALL
16535 133 16517
16536 64 AND
16537 3 INC BC
16538 3 INC BC
16539 3 INC BC
16540 235 FOR EX DE,HL
16541 113 ? LD (HL),C
16542 35 7 INC HL
16543 112 ? LD (HL),B
16544 11 " DEC BC
16545 11 " DEC BC
16546 11 " DEC BC
16547 62 Y LD A,
16548 0 0
16549 35 7 INC HL
16550 35 7 INC HL
16551 54 0 LD (HL)
16552 27 " 27
16553 11 " DEC BC
16554 184 S CP B
16555 32 4 JR NZ,
16556 249 RAND -7
16557 185 S CP C
16558 32 4 JR NZ,
16559 246 PLOT -10
16560 201 TAN RET

```

FIG 8-1. BUILDING A MEGA-REM

1. There must be no line number 11.
2. No line should contain exactly 9 nulls. (They may be as long as you can conveniently handle - or as short as 1 byte.)
3. No line should contain a quotation mark, except for the last one.
4. The lowest-numbered line should be number 2 or above.

When you think you have the number of nulls that you want, then PRINT USR 16517 (note the new number) will tell you how many nulls will be in the final MEGA-REM. You can adjust that number by adding or deleting lines, or adding or deleting nulls within any of the lines (always observing the provisos listed above). When the number is right, you can consolidate them all by RAND USR 16514.

And while you're wondering what to do with that huge block of nulls, let's go back to the lesson.

ZX81 BASIC lets you define a virtually unlimited number of variables--anything that starts with a letter and contains only letters and numbers. So a variable in BASIC could be A, AARDVARK, A1,...on through ZYMURGY, ZZZ, and beyond.

Not so with machine code. We only have seven variables to work with (well, we do have a few others, that we'll hold in reserve for the the moment), and we can't be arbitrary about their names. Somebody has already given them a rather prosaic series of names: A, B, C, D, E, H, and L. At least they showed a little bit of creativity--the last two could have been F and G. (As we mentioned earlier, the H and L stand for High and Low, in some contexts.)

The "in" group of machine codists don't talk about them as variables--they speak in terms of "registers", which are "loaded", instead of variables which are assigned a value. Since this term is used by the highbrows, we'll use it, alongside our BASIC analogies. And then, after a while we'll drop the analogies. After you've seen the two systems side by side a couple of times, you'll start to feel at home with the new convention. Sort of like watching an English-language movie with foreign subtitles.

We already know how to set two variables:

LET B = or LD B, is 6 (06h)

LET C = or LD C, is 14 (0Eh)

The symbols LD B, and LD C, which of course stand for LOAD B and LOAD C, are known as MNEMONICS (and that's a great letter for your phonetic alphabet--just think: P as in pneumonia, M as in mnemonic...). The comma is a part of the mnemonic, and serves to separate the instruction from its argument.

The seven registers can be loaded by means of the instructions given in Table 8-1. You'll also find these, and the rest of the Z80's instruction set, in the back of the ZX81 manual, under the section on the Character Set.

So now you can load any of several registers with a number from 0 to 255. Only you can't look at any of these registers except B and C. Remember, all you can get the computer to say is "256*B+C=..."

MNEMONIC	MEANING	INSTRUCTION
LD A,N *	LET A=N	62
LD B,N *	LET B=N	6
LD C,N *	LET C=N	14
LD D,N *	LET D=N	22
LD E,N *	LET E=N	30
LD H,N *	LET H=N	38
LD L,N *	LET L=N	46
* EACH INSTRUCTION REQUIRES A SINGLE BYTE OF DATA		

TABLE 8-1. LOADING SINGLE-BYTE REGISTERS

Obviously, we have to find some way to shift data from one register to another. And that's where Table 8-2 comes in.

LD R1,R2		(LET R1=R2)							
(LET	=	)	A	B	C	D	E	H	L

A,			127	120	121	122	123	124	125
B,			71	64	65	66	67	68	69
C,			79	72	73	74	75	76	77
D,			87	80	81	82	83	84	85
E,			95	88	89	90	91	92	93
H,			103	96	97	98	99	100	101
L,			111	104	105	106	107	108	109

TABLE 8-2. SHIFTING DATA BETWEEN REGISTERS

It's a little puzzling, by the way, why they provide instructions that will LET A=A, LET B=B, and so on, but there they are.

So let's try out what we know. Enter the program of Figure 8-2, using the loader program. Did you get 1? Now try Figure 8-3. Your output should be 256.

OPERATION	MNEMONIC	INSTRUCTION
LET A=	LD A,	62.
1	1	1.
LET B=	LD B,	6.
0	0	0.
LET C=A	LD C,A	79.
RETURN	RET	201.
(PRINT	PRINT USR 16514	
256*B+C)		

FIGURE 8-2. LOADING REGISTERS

OPERATION	MNEMONIC	INSTRUCTION
LET H=1	LD H,1	38.1.
LET L=0	LD L,0	46.0.
LET B=H	LD B,H	68.
LET C=L	LD C,L	77.
RETURN	RET	201.
(PRINT	PRINT	USR 16514
256*B+C)		

FIGURE 8-3. LOADING REGISTERS
(CONTINUED)

Try several other combinations: LET D=1 (LD D,1) and LET E=0 (LD E,0), for instance. Shift them into B and C respectively, and see if you come out with 257. Then with the same values in D and E, shift them into C and B respectively, and see if you come out with 1.

If you want to get really fancy, try passing a variable from E to H before bringing it back to B or C. Do the same with the other register of the BC pair. Sort of like Tinker to Evers to Chance. Or, in 1984, like Sandberg to Bowa to Durham. Remember, though, you have to end up with a defined value for B and C in order to get any meaningful output.

And remember, as we learned in Chapter 6 (SWN 2:6), if you don't assign any value to BC, it just takes on the starting address of the USR routine.

Doubling Up

Fine, you say, but what if I want to use numbers bigger than 255? Well, as the candidate says, I'm glad you asked that question. And you may have already guessed the answer--use 256-imal. Remember that BC will handle any number up to 65535--one less than 256*256. So why not treat other register combinations as pairs?

Actually, you've already been doing that. Remember when you LET H=1 and LET L=0? That was the same as Loading HL with 256. To pair two registers, all

MNEMONIC	MEANING	INSTRUCTION
LD BC,NN	* LET BC=NN	1
LD DE,NN	* LET DE=NN	17
LD HL,NN	* LET HL=NN	33
* NOTE THAT 2 BYTES OF DATA MUST BE SUPPLIED, LEAST SIGNIFICANT BYTE FIRST		

TABLE 8-3. LOADING REGISTER PAIRS

you have to do is load one of them with all the 256's in your number, and load the other register with what's left over. Sort of reminds me of the man who went to a hotel for a rest and a change, but Room Service got all the change, while the hotel got all the rest.

Or, in mathematical language, to Load H with 258, you

```
LET H=INT(258/256)  =1
and LET L=258-256*H  =2
```

Three pairs of the Z80's registers are set up to work AS PAIRS. Look at Table 8-3. Unfortunately, Zilog, the company that designed the chip, didn't provide any way for shifting data from one pair to the other--you have to move data one register at a time.

Remember, a register pair must be loaded with two bytes of data, in 256-imal. If you tell the computer to load a register pair, you've implicitly promised it two bytes of data to chomp on. If you don't provide two of them, the computer will take a byte out of the next instruction, and your program will be in real trouble. Table 8-3 reminds you of this fact by using the symbol NN to represent the two-byte data unit.

Double Trouble

Let's use the loader program (also in SWN 2:6-ed.) to enter

```
100 LET P$="1.1.0.201."
```

LET B=1, and LET C=0, no? Try it, and see. You'd expect the printout of BC to give you 256, wouldn't you? So what do you get? 1! Try

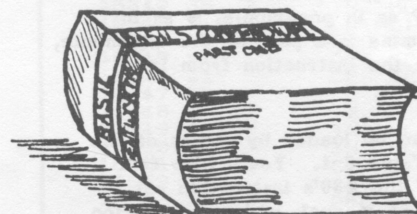
```
100 LET P$="1.0.1.201."
```

And there's the 256 you wanted.

When you Load a two-byte register, the computer wants the Least Significant Byte first, and the Most Significant Byte second. Say that over and over again to yourself. If it'll help you to think of loading LH instead of HL, remember it that way. Or you might remember that the bytes are loaded in alphabetical order: LSB followed by MSB.

Remember it any way you want--any mnemonic that works is a good mnemonic. But one way or another, remember it.

That's all for now. There will be more.



Mike Trivisonno
BASIS COMPENDIUM...
PART ONE

A TS2068 MUSIC MAKER

Ronald Rissel

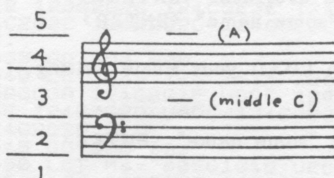
The Timex/Sinclair 2068, with its built-in sound chip is capable of producing sound and music in three parts. The program included here will allow you to create, enter, and play music in three-part harmony. It doesn't have the visual appeal of the famous Music Construction Set (made for lesser computers), but it does allow you to vary the volume of the sound within a musical composition, which the aforementioned program lacks. The Music Editing Program, given in listing 1, includes a couple of machine language routines stored in lines 1200 through 1260.

Typing Music In

Music is entered in a straight-forward numerical fashion directly in the program between the lines numbered 500 and 895. It may be typed in directly or merged in from tape (more about this later). In general, you will have a line that defines a measure or two of music (in the character string M\$), followed by a line that calls the encoding routine at line 900. Look at line 510 in listing 1. The first group of letters (VFED) is called a volume group and it sets the volume of the three channels. A volume group starts off with the letter V followed by three hexadecimal digits (0-F) with 0 being minimum volume (essentially off) and F representing maximum volume. Once the volume of the three channels is set, it will remain until changed.

The next thing you will notice in line 510 (243C3E3G) is called a note group. The first two numbers (the 24) is the length of the note group. A 24 represents a quarter note. Twice that, or 48 would represent a half note, and 96 would represent a whole note. Likewise, a 12 represents an eighth note and a 06 (must have 2 digits) represents a 16th note. Following the note length, you will have 3 sets of note specifiers, each of which consists of an octave specifier (the 3), followed by the actual note itself (A through G), and possibly followed by either a "b" or a "#" for either a flat or a sharp (natural notes don't need to be followed by anything). See figure 1 for the octave specifier.

Figure 1
Octave Numbers



Further along in line 510 you will see a group consisting of an R04. This is called a repeat group and it is a short cut method of representing the same notes as before with a possibly different length. This is useful when varying the volume of a note or group of notes, or for representing dotted notes or tied notes.

LISTING 1: THE MUSIC EDITING PROGRAM

```

1 REM MUSIC EDITING PROGRAM
2 REM Music starts at line 50
0) Adjust DIM L$(*) as needed.
10 CLEAR 65100
20 DIM L$(3000): DIM M$(200):
DIM C$(1)
30 GO TO 1000
40 PAUSE 20: SOUND 7,56
45 LET HL=6+PEEK 23627+256*PEE
K 23528: REM Get start of L$
50 POKE 65109,HL-256*INT (HL/2
56): POKE 65110,INT (HL/256): RE
M Poke in value for CHRPTR
55 POKE 65111,L-256*INT (L/256
): POKE 65112,INT (L/256): REM P
oke in value for NCHARS
60 PAUSE USR 65115
65 IF (PEEK 65111+PEEK 65112)>
0 THEN GO TO 60
70 SOUND 7,63:8,0:9,0:10,0
75 INPUT "A)gain or Q)uit? ";C
$
80 IF C$="Q" OR C$="q" THEN ST
OP
85 GO TO 40
100 REM CALCULATE TIME TO PAUSE
110 LET L=L+1: LET TM=INT (1.5*
T*VAL M$(M TO M+1))
120 IF NOT TM THEN LET TM=1
130 LET L$(L)=CHR$ TM: LET M=M+
2
140 RETURN
150 REM NOTE GROUPS
160 GO SUB 100
170 FOR I=1 TO 3: REM 3 Channel
S
180 LET OCTAVE=12*VAL M$(M)-12
190 LET M=M+1: LET C$=M$(M)
200 IF C$<"A" OR C$>"G" THEN BE
EP .2,20: PRINT "**ERROR** Wrong
note received at M = ";M: ";
C$: STOP
210 IF C$="A" THEN LET NOTE=0
211 IF C$="B" THEN LET NOTE=2
212 IF C$="C" THEN LET NOTE=3
213 IF C$="D" THEN LET NOTE=5
214 IF C$="E" THEN LET NOTE=7
215 IF C$="F" THEN LET NOTE=8
216 IF C$="G" THEN LET NOTE=10
217 IF M$(M+1)="#" THEN LET NOT
E=NOTE+1: LET M=M+1
218 IF M$(M+1)="b" THEN LET NOT
E=NOTE-1: LET M=M+1
220 POKE 65106,OCTAVE+NOTE: REM
INPUT NO.
230 RANDOMIZE USR 65200
240 LET L$(L+1)=CHR$ (PEEK 6510
6): REM FINE NO.
250 LET L=L+2: LET M=M+1
260 LET L$(L)=CHR$ (PEEK 65107)
: REM COARSE NO.
270 NEXT I
280 LET N$=L$(L-5 TO L): REM Sa
ve notes for possible repeat gro
up
290 GO TO 950
300 REM VOLUME GROUPS
310 LET L=L+1: LET M=M+1: LET L
$(L)=CHR$ 255
320 FOR I=1 TO 3
330 LET V=CODE M$(M)-48: IF V>9
THEN LET V=V-7
340 LET L=L+1: LET M=M+1: LET L
$(L)=CHR$ V
350 NEXT I
360 GO TO 950
400 REM REPEAT GROUPS
410 LET M=M+1: GO SUB 100
420 LET L=L+6: LET L$(L-5 TO L)
=N$
430 GO TO 950
500 REM VFED 243C3E3G
510 LET M$="VFED 243C3E3G 243C3
F4R 123C3E3G VEDC R04 UCBA R04 V
A98 R04 U876 R04 U654 R04 U432 R
04 U211 R04."
515 GO SUB 900

```

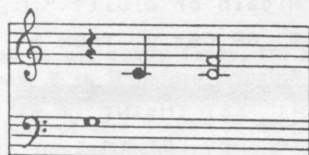
Finally, at the end of line 510, you will notice a period just before the ending quotes. Each line of music must end with a period. (If you forget, you will get an error #3, Subscript Error in line 950, when you try to run the program later.)

If you only have one or two notes to play at a time, use a volume group to turn the other notes off; the note values you give them then doesn't matter. As an alternative, you could leave the volume the same, in which case you can give them the same note value as the one (or two) notes you are playing. Another problem that will come up is what to do when the notes you want to play are not the same in length, such as when the bass staff has a whole note and the treble staff has 4 quarter notes. In this case, you need to convert the whole note into 4 "tied" quarter notes. Figure 2 illustrates some of these problems, and would be encoded thusly:

"242G2G2G 242G2G3C 482G3C3E."

With a little practice, you'll be writing music fairly quickly.

Figure 2



Running the Program

First make sure you save and verify your entered program before you try to run it! Now with some music entered between lines 500 and 885 (the listing includes a couple of measures of sample music), type RUN. The program will take about 10 seconds to poke in the machine code portions, then it will ask you to "Input Tempo" with suggested values of .5 to 1.5. Smaller numbers will make the music play faster while larger numbers will make it play slower. For now, enter a 1. The program will then say "Please wait...". It is now converting your music code into numbers that the computer can understand. Each call to the subroutine at 900 will draw one more dot on the screen, so if you have any errors, you can count dots (minus the first 3) to determine which line has the error in it. Usually the error will be in a note group; you may have typed a 6 for the length when it should be a 06, or you specified a note in small letters when they should be in capitals. Also, if you entered a long piece of music, you may get an error 3--Subscript Wrong. In this case, you need to enlarge the dimension of L\$ in line 20. If the computer completely crashes during any part of this (or the next part when you actually play the music), you've made a mistake somewhere in the data statements in lines 1200 to 1260. Re-load your program, find and correct the error(s), re-save your program, and try running it again.

Assuming you've got all the bugs out, the computer will finally say, "Music is now ready to be played. Hit ENTER." Go ahead, hit ENTER, and sit back and enjoy your masterpiece!

```
520 LET M$="U000 0638b3Eb3G# UF
DC 2438b3Eb3G# 2438b3Eb3G U000."
525 GO SUB 900
530 LET M$="243G2G1G UFDC R12 U
000 003G2E1G UFDC R06 U0DC R00 U
FEC R06 U00C 003G2G1G UFDC R12 1
2383A1G 123G3B1G 123G3D1G."
535 GO SUB 900
890 BEEP .2,20: PRINT : PRINT
892 PRINT "Music is now ready t
o be played.Hit ENTER."
894 INPUT C$
896 CLS : GO TO 40
900 LET M=1: REM Index into M$
950 LET C$=M$(M)
960 IF C$>="0" AND C$<="9" THEN
GO TO 150
965 IF C$="R" THEN GO TO 400
970 IF C$="U" THEN GO TO 300
975 IF C$=" " THEN LET M=M+1: G
O TO 950
980 IF C$="." THEN PRINT ".":
RETURN
990 BEEP .2,20: PRINT "**ERROR*
*AT M= ";M;" : C$: STOP
1000 PRINT "Loading machine code
portion..."
1010 FOR A=65115 TO 65355
1020 READ D: POKE A,D
1030 NEXT A
1100 INPUT "Input Tempo (try .5
to 1.5): ";T
1110 CLS : PRINT "Tempo = ";T
1120 PRINT : PRINT "Please wait.
..."
1130 LET L=0
1140 GO TO 500
1200 DATA 42,85,254,205,164,254,
254,255,32,11,62,8,205,147,254,2
54,11,56,249,24,19,50,89,254,151
,205,147,254,254,6
1210 DATA 56,249,58,89,254,6,0,7
9,24,12,58,87,254,183,32,213,58,
88,254,183,32,207,34,85,254,201
1220 DATA 1,245,0,237,121,245,20
5,164,254,1,245,0,237,121,241,60
,201,126,35,237,75,87,254,11,237
,67,87,254,201
1230 DATA 33,204,254,205,195,254
,50,83,254,33,12,255,205,195,254
,50,84,254,201,58,82,254,6,0,79,
9,126,201
1240 DATA 7,7,8,6,6,5,5,4,4,4,
4,3,3,3,3,3,2,2,2,2,2,1,1,1,
1,1,1,1,1,1,1,1,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0
1250 DATA 196,85,235,136,42,209,
126,47,228,158,92,29,226,170,117
,68,21,232,191,151,114,79,46,14,
241,213,186,162,138,116
1260 DATA 95,75,57,39,23,7,248,2
34,221,209,197,186,175,165,156,1
47,139,131,124,117,110,104,96,93
,87,82,78,73,69,65,62,58,55,52
```

Saving Your Music

Once you have your music entered and sounding the way you like, you'll want to save it. The simplest, most convenient way would be to save the whole program, but if you're worried about saving tape, there is another way. Assuming that you have already saved the basic Music Editing Program, type:

```
DELETE 1,490 <ENTER>
DELETE 890,1260 <ENTER>
SAVE "some name" <ENTER>
```

Then later on, you can LOAD your Music Editing Program, and then type:

```
MERGE "some name" <ENTER>
```

and your program will be ready to run! Then if you want to merge in a different music score, type:

```
DELETE 500,885 <ENTER>
```

to delete your old music (VERY IMPORTANT!), then type:

```
MERGE "new name" <ENTER>
```


where "new name" is the filename for another music file.

If you would like to save yourself some typing, and get several songs as well, send a check for \$6.00 to: Ronald Rissel 2967 Lynx Lake Drive Prescott Vly., AZ 86314

I'll send you a tape containing the Music Editing Program, three songs already encoded and ready to be merged, along with their corresponding sheet music, and complete instructions. I'll also be glad to answer any questions that you may have, if you'll write to me at the above address.

LISTING 2: SAMPLE MUSIC CODE
from "The Music Box Dancer"
by Frank Mills

```
500 REM
510 FOR J=1 TO 3
515 REM (Included in case you
      forgot to type DELETE 500,885
      before merging this music.)
520 LET M$="V9E0 123C3C3C 123E3
E3E 123G3G3G 123E3E3E 124C4C4C 1
23G3G3G 123E3E3E 123G3G3G."
525 GO SUB 900
530 NEXT J
535 REM (another safeguard)
540 LET M$="123C3C3C 123E3E3E 1
23G3G3G 123E3E3E 124C4C4C 123G3G
3G 123E3E3E VED0 124C3G3G."
545 GO SUB 900
550 LET M$="V7C0 004C3C3C VED0
R12 123G3E3E 124C3G3G 124E4C4C 1
24C3E3E 124E3G3G 124G4C4C 124C3G
3G."
555 GO SUB 900
560 LET M$="125C3F3F 125B4A4A 1
25A4C4C 124G4F4F VDD0 124G3C3C U
CD0 124G3E3E V8D0 124G3G3G V7D0
124G4C4C."
565 GO SUB 900
570 LET M$="VED0 124G2G2G 124F3
B3B 124D3D3D 124B3G3G 123G3B3B 1
24B3D3D 124D3G3G 124F3D3D."
575 GO SUB 900
580 LET M$="124E3C3C 124C3E3E 1
25A4C4C 124G3G3G VDD0 124G3E3E U
CD0 124G3G3G V8D0 124G4C4C 004C3
G3G VED0 R12."
585 GO SUB 900
590 LET M$="V7C0 004C3C3C VED0
R12 123G3E3E 124C3G3G 124E4C4C 1
24C3E3E 124E3G3G 124G4C4C 124C3G
3G."
595 GO SUB 900
600 LET M$="125C3F3F 125B4A4A 1
25A4C4C 124G4F4F VDD0 124G3C3C U
CD0 124G3E3E V8D0 124G3G3G V7D0
124G4C4C."
605 GO SUB 900
610 LET M$="VED0 124G2G2G 124F3
B3B 124D3D3D 124B3G3G 123G3B3B 1
24B3D3D 124D3G3G 124B3D3D."
615 GO SUB 900
620 LET M$="124C3C3C 123G3E3E 1
24E3G3G 124C4C4C VDD0 124C3E3E U
CD0 124C3G3G V8D0 124C4C4C 004C2
G2G VED0 R12."
625 GO SUB 900
630 LET M$="V800 004C2C2C VEDC
R12 123G2G2C 124C3E3C 124E3E3C 1
24C2G2C 124E2G2C 124G3E3C 124C2C
2C."
635 GO SUB 900
640 LET M$="125C1F1F 125B2F1F 1
25A3C3A 124G2G1F VDDC 124G2C2C U
CDC 124G2G2G V8DC 124G3E3C V7D0
124G2D2D."
645 GO SUB 900
650 LET M$="005G1G1G VEDC R12 1
25F2G1G 125D3D3B 125B3D3B 124G2G
1G 125B2G1G 125D3D3B 125F2G1G."
655 GO SUB 900
660 LET M$="125E2C2C 125C2G2C 1
26A3E3C 125G3E3C VDDC 125G2G1G U
C8C R00 VDC R12 V7CB 125G3E3C 0
04C2G1C VEDC R12."
665 GO SUB 900
670 LET M$="V800 004C2C2C VEDC
R12 123G2G2C 124C3E3C 124E3E3C 1
24C2G2C 124E2G2C 124G3E3C 124C2C
2C."
675 GO SUB 900
```

```
680 LET M$="125C1F1F 125B2F1F 1
25A3C3A 124G2G1F VDDC 124G2C2C U
CDC 124G2G2G V8DC 124G3E3C V7D0
124G2D2D."
685 GO SUB 900
690 LET M$="005G1G1G VEDC R12 1
25F2G1G 125D3D3B 125B3D3B 124G2G
1G 125B2G1G 125D3D3B 125B2G1G."
695 GO SUB 900
700 LET M$="125C2C2C 124G2G2C 1
25E3E3C 125C3E3C 125C2G1C V7A9 R
00 VEDC R11 V7CB 005C3E3C VEDC R
11 V7CB 005C2G1C VEDC R11."
705 GO SUB 900
710 LET M$="025C1F1F 025E5C1F 0
26A5E1F 066C6A1F 126A2F1F 125F3C
3A 125C3C3A 125A2F1F 125C2F1F 12
5F3C3A 126A2G1F."
715 GO SUB 900
720 LET M$="125G2C2C 125C2G2C 1
26A3E3C 125G3E3C VDDC 125G2G2C U
E88 R00 VDDC R12 V8DC 125G3E3C U
7DC 125G2G2D."
725 GO SUB 900
730 LET M$="VEDC 024G1G1G 025B4
G1G 025D5B1G VDC 065G5D1G 125F2
G1G 125D3D3B 125B3D3B 124G2G1G 1
25B3B1G 125D3D3B 125F2G1G."
735 GO SUB 900
740 LET M$="125E2C2C 125C3C2C 1
26A3E3C 125G3E3C 125G3C2C 125E3C
2C V7BA 005E3G3C VEDC R11 V7BA 0
05E2F2C VEDC R11."
745 GO SUB 900
750 LET M$="025C1F1F 025E5C1F 0
26A5E1F 066C6A1F 126A2F1F 125F3C
3A 125C3C3A 125A2F1F 125C2F1F 12
5F3C3A 126A2G1F."
755 GO SUB 900
760 LET M$="125G2C2C 125C2G2C 1
26A3E3C 125G3E3C VDDC 125G2G2C U
EDC 125C2G1C V8CB 005C3E3C VEDC
R12 V8CB 005C2G2C VFC R12."
765 GO SUB 900
770 LET M$="024G1G1G 025B4G1G 0
25D5B1G 065G5D1G 125F2G1G 125D3D
3B 125B3D3B 124G2G1G 125B2G1G 12
5D3D3B 125B2G1G."
775 GO SUB 900
780 LET M$="135C2C2C 134G2G2C 1
45E3E3C 145C2G2C VDDC 155C3E3C U
DCB R06 VCB R06 VBA9 R06 VA98 R
06 V987 R02 V876 R01 V765 R01 V5
43 R01."
785 GO SUB 900
```

TYD☆BYTS

Fun With HOT-Z II

Hot Z II uses a fascinating "trick" for getting back into itself if you should stumble into the error restart, RST 08. This is done by jimmying the ER_SP system variable. For a graphic demonstration, type 0676 (disassembly mode), then go to edit mode and run the code starting at this address (shift R). Dumps you right back into BASIC, doesn't it?

Now enter any command with a syntax error. For example, LET, PLOT, etc. without a valid argument. Wow!

Want to see how it's done? Enter 5800 (cold boot initialization), then flip to the next page using ENTER. Change the command, LD HL,HZET to LD HL,CHOO. You can now stumble into an error restart without messing up your names files, etc. Can anyone out there work out a two-way "door" between BASIC and machine-code using this gimmick? -fn

TMS9918A VIDEO UPGRADE FOR THE TS1000

Oliger/McMullin/Nachbaur

A Follow-up

As can be expected with any major hardware upgrade, there are a few conflicts when the Oliger video system (see SWN #2:1 and 2:2) is used in conjunction with other equipment. Sadly, some manufacturers shed little light on what has to be done to make their add-ons compatible with other peripherals. This is not the case with the John Oliger Company. In the process of building up my Super-ZX, I found its illustrious proprietor most helpful in getting my system working harmoniously. Most of the information in this article was supplied by John in response to my many letters and phone calls, and since his schedule is already very busy I (Fred N.) agreed to "ghost-write" this article to summarize the fixes we and others have come up with to date.

ROM Chip Select

The most common conflict involves devices that use the ROMCS NOT line to disable the ROM. There are two reasons that a device would want to do this. The first is that the ZX/TS doesn't fully decode the ROM address space, with the result that the ROM "echoes" in the 8-16K range. Any memory addition (64K packs, Hunter board) that employs this region of memory must force ROMCS NOT low when A14, A15 and MREQ NOT are low, and A13 is high (8-16K is being addressed). This usage of ROMCS NOT DOESN'T affect operation with the video upgrade, since its purpose is only to supply decoding absent in the Sinclair logic chip. This decoding is duplicated on video board "B" for EPROM "a" using gates in U1 and U2.

The other reason for using ROMCS NOT is to actually OVERLAY or MASK the ROM when certain addresses are paged. Any device that contains its own code to replace part of the ROM code has to turn off the ROM under these conditions. Devices that do this include the Memotech Centronics interface and the Compusa disk drive, quite probably several others. A problem arises with the video upgrade, because during normal operation you're not even using the resident ROM; the modified ROM code in EPROM "a" is used instead.

Therefore, the ROMCS NOT line from your outside device has to be patched in so that it controls the chip select on EPROM "a" instead of the original ROM.

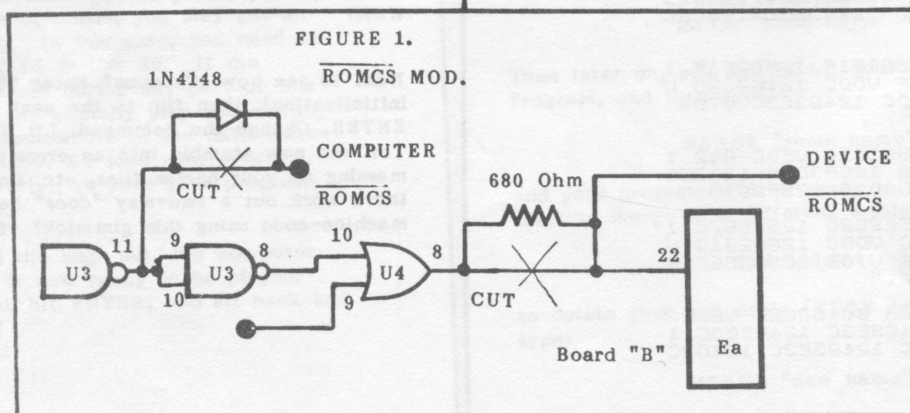
Figure 1 shows how to accomplish this. As you can see, the solution is very easy, and involves only video board "B":

- 1: Cut the line from the junction of U3 pin 9,10 and 11 to ROMCS NOT. Bridge the gap with a 1N4148 diode, cathode (banded) end towards computer ROMCS NOT.
- 2: Cut the trace between U4 pin 8 and Ea pin 22. Bridge the gap with a 680 ohm resistor.
- 3: Disconnect your outside device's (e.g. Memotech interface) ROMCS NOT line, and connect it to Ea pin 22 instead.

If you are using John's expansion board, the required re-route of the interface's ROMCS NOT line can be accomplished easily with a trace cut, and a jumper on board "B" to one of the unused pins on the expansion board (e.g. 24B), which we could call EaCS NOT. Note, however, that you cannot simply unplug the video boards to return to the normal Sinclair video system and still have your other peripherals work normally. The solution is to provide a switch to toggle between modes.

Switchable Video

Though the Oliger TI video system is vastly superior for most programs, there are some programs (especially machine-code games) that run far too fast to be usable. Or you might want to run Software Farm's "software-only hi-res" games or similar programs (which don't work with Oliger video). For such software it would be convenient to have a single switch to toggle between normal Sinclair video and Oliger TI video. This is especially vital



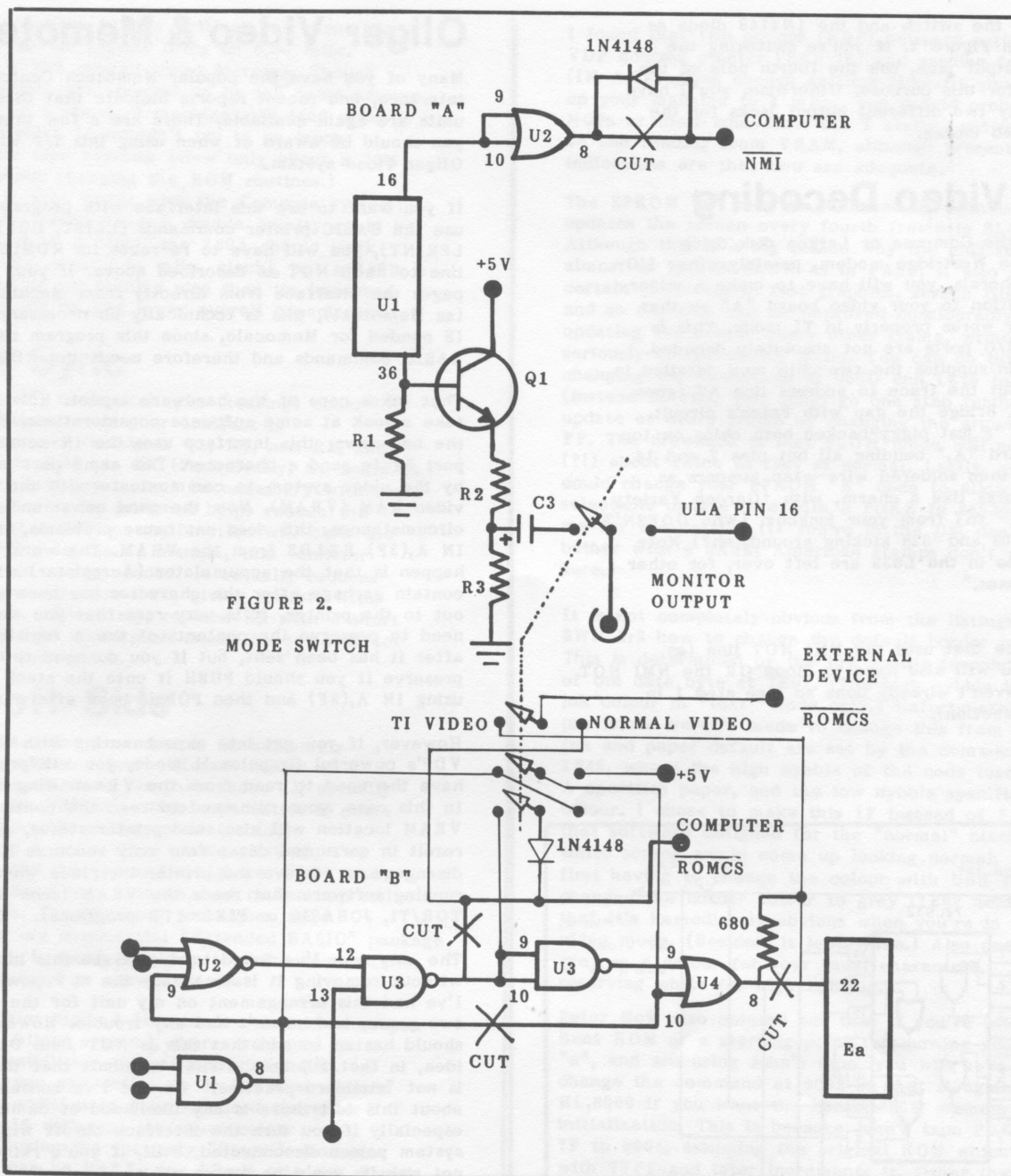


FIGURE 2.
MODE SWITCH

If you, like me, have built your system into a single enclosure, and cannot easily unplug the video boards.

John worked out a way to do this using a DPDT switch. If you also have to switch the out-going ROMCS NOT line as above, you'll need a third pole. You could go one step further yet and add a fourth pole to switch the video output jack between the TMS9918A output and the normal ULA pin 16 output (or video driver, e.g. as detailed in SWN 1).

Since the mechanics of mounting the switch will vary according to your particular installation, I'll leave the mounting details up to you. Try to mount the switch as close to the video boards as is practical. The electronics mods are quite simple, and are diagrammed in Figure 2. The changes are:

- 1: Cut the trace going to pin 8 of U2 on board "A", and bridge the gap with a 1N4148 diode, cathode end towards U2 pin 8 (see Figure 3).

This allows the normal video system to continue using the NMI NOT line when the TI video is disengaged. This is the only change required on board "A" unless you decided to also switch the video line.

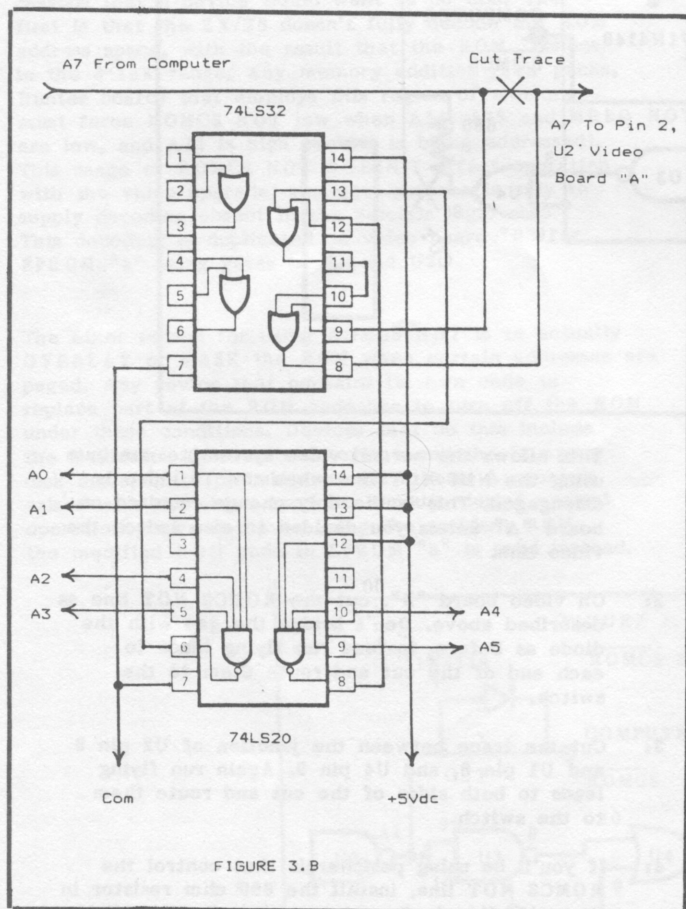
- 2: On video board "B", cut the ROMCS NOT line as described above. Don't bridge the gap with the diode as before, instead run flying leads to each end of the cut and route them to the switch.
- 3: Cut the trace between the junction of U2 pin 9 and U1 pin 8, and U4 pin 9. Again run flying leads to both sides of the cut and route them to the switch.
- 4: If you'll be using peripherals that control the ROMCS NOT line, install the 680 ohm resistor in line with U4 pin 8, and route the EaCS NOT line from Ea pin 22 to the switch.

- 5: Wire up the switch and the 1N4148 diode as shown in Figure 2. If you're switching the video output jack, use the fourth pole of the switch for this purpose. Otherwise, you'll have to supply two different output jacks for the two video modes.

Oliger Video Decoding

If you have the Compusa or Larken disk drive controller, the Westridge modem, possibly other I/O-mapped peripherals, you will have to make a minor decoding addition to your video board "A" so that the controller works properly in TI mode. This is because the I/O ports are not absolutely decoded. Peter McMullin supplies the two-chip mod detailed in Figure 3.B. Cut the trace to address line A7 (goes to U2 pin 7). Bridge the gap with Peter's circuit. Peter writes, "I just piggy-backed both chips on top of U2 on board "A," bending all but pins 7 and 14 straight out, then soldered wire-wrap jumpers as necessary. Works like a charm, with †Garden Variety LS' [or HCT - fn] from your junkbox. (Who DOESN'T have some '20s and '32s kicking around, eh?) Note that two gates in the LS32 are left over, for other hacking purposes."

For any device that uses the NMI NOT line (as Compusa), you will also have to diode OR the NMI NOT line if you haven't already done so (see step 1 in the previous section).



Oliger Video & Memotech...

Many of you have the popular Memotech Centronics interface, and recent reports indicate that these units are again available. There are a few things you should be aware of when using this I/F with the Oliger video system.

If you want to use this interface with programs that use the BASIC printer commands (LLIST, COPY, LPRINT), you will have to re-route its ROMCS NOT line to EaCS NOT as described above. If your program pages the interface from directly from machine-code (as Memotext), this is technically unnecessary. It IS needed for Memocalc, since this program uses the BASIC commands and therefore needs the ROM overlay.

That takes care of the hardware aspect. Now let's take a look at some software considerations. First the bad news: this interface uses the IN command at port 3F to send a character. This same port is used by the video system to communicate with the VDP's video RAM (VRAM). Now the good news: under normal circumstances, this does not cause problems, since IN A,(3F) READS from the VRAM. The worst that can happen is that the accumulator (A register) will contain garbage after the character has been shipped out to the printer. It is very rare that you would need to preserve the contents of the A register after it has been sent, but if you do need to preserve it you should PUSH it onto the stack before using IN A,(3F) and then POP it back afterwards.

However, if you get into experimenting with the VDP's powerful Graphics II mode, you will probably have the need to read from the VRAM using IN A,(3F). In this case, your command to read the contents of a VRAM location will also read printer status, and result in corrupted data. Your only recourse is to disengage or remove the printer interface when running software that reads the VRAM (such as my WATOR/TI, JOBASIC or PIXL-ATR programs).

The only way I've found to deactivate this interface without removing it is to switch the +5V power rail. I've had this arrangement on my unit for the past two years, and haven't had any trouble. However, I should hasten to add that this is NOT John Oliger's idea, in fact I'll be the first to admit that this is not "standard practice." No one I've consulted about this felt there is any likelihood of damage, especially if you turn the interface on/off with system power disconnected. Still, if you'd rather not risk it, you'd be better off to physically remove the interface when its presence causes software conflicts.

...and Aerco

Similarly, the Aerco printer interface uses port 7F, which is also used by the video system to communicate with the VDP. This is the only known "irreconcilable difference" at this time. If you're ambitious, you might consider re-mapping one or the other device. Alternately, consider a different interface for use with Oliger video.

Reports from the field indicate that no changes are required to make the Oliger video system work properly with the Aerco disk drive controller.

Speed it Up

With the Oliger video system, you eliminate dependence on the ULA logic chip as far as video generation is concerned. As a result, you can install a faster crystal and get your machine

running even faster. At this writing, Tom Bent has successfully run his system using an 8.00 MHz. crystal, for a 23% overall speed increase. Though the tape routines are speeded up by the same amount, tapes saved on a standard machine still load using the faster crystal. (I wouldn't try to go much faster, as the tape routines allow only about a 30% increase without changing the ROM routines.) Amazingly, it even works with the Compusa system! (Single density recommended; double density could be iffy.) In principle at least, the Z80A should work with an 8 MHz. crystal (4 MHz. clock), but a Z80B is recommended, if for nothing else than its improved load handling ability and lower overall capacitance.

Get in Sync

If you are losing horizontal sync (tearing) under certain conditions, or if your contrast is inadequate, remove transistor Q1, R2, and R3, and connect C1+ to VDP pin 36. (Technically, C1 could be omitted also, and may in fact be necessary with some monitors. However, its presence helps reduce the possibility that a monitor malfunction will blow the VDP). According to the TI manual, this "emitter follower" is not really needed. It actually reduces the video signal by some 30%, giving trouble with some monitors. Alternately, replace R2 with a short circuit, which will accomplish much the same effect as the direct connection.

The Soft Side

The TMS9918A video processor is an extremely flexible chip, allowing four different modes of operation. One of these, Graphics mode II, actually outperforms the TS2068's video capabilities in some ways (e.g. eight times the vertical colour resolution, and a much more stable picture). I've successfully used this mode for high-res PLOT and DRAW, 64-column text screens, and rather impressive colour displays. Though these techniques were developed for my commercial "Extended BASIC" package for this board, I would be happy to share some of the details if there is enough interest.

Even in Graphics mode I (the mode selected when you initialize the Olliger system), there are some interesting possibilities not previously reported. For instance, you can define up to 128 user-defined characters in addition to your standard character set, in up to 16 different ink/paper colour combinations, using the formerly "illegal" character codes. These can be POKed (or LDed) into the display file directly, for spectacular "space invaders" and other such creatures.

A third mode allows individual colour control over low-res block graphics (comparable to standard PLOT/UNPLOT) might be useful in certain graphic applications.

The fourth mode ("Text") allows 40 column screens but is actually of limited usefulness, since up to 64 columns can be implemented in Graphics mode II, with far superior colour capabilities. Still, there are applications where this mode can be used to advantage. Sean Wenzel is currently working on a modem terminal program using this mode because of its higher speed (as compared to Mode II 64-column) and because most BBS's use a 40-column format.

Now for some practical software tips when using this device. First is the matter of timing; the TMS9918A can't quite keep up with even a 3.25 MHz. Z80 under certain circumstances. In general, sending data TO the VRAM (using OUT 3F,A) is not troublesome, but READING (using IN A,3F) seems to be a little balky.

I found that two no-ops (NOP) between setting the VDP address pointer (RST 38) and reading from VRAM (IN A,3F) gives enough settling time. If you speed up your machine to 4 MHz., you should probably use 3 NOPs or their equivalent (12 T states) between RST 38 and reading from VRAM, although present indications are that two are adequate.

The EPROM "a" code as published by John Olliger only updates the screen every fourth frame in SLOW mode. Although this is a big reason why the unit runs almost as fast in SLOW as in FAST, it also causes a certain jumpiness in fast graphics, screen fills, and so on. This can be improved considerably by updating the screen every other frame, without seriously compromising speed. You can do this by changing the commands at 006D and 1E76 to LD A,FE (instead of FC). You could go one step further and update at every frame by changing the argument to FF. The drawback is that SLOW mode will now be only (?) about twice as fast as normal. Lastly, you could change your EPROM "A" code to allow a selectable update rate with a POKE to 65535. Contact me for the details. (If you're in the US, don't bother with a SASE; American stamps don't work from here.)

It is not completely obvious from the listings in SWN 2:2 how to change the default border colour. This is determined by the low nybble (second digit) of the data byte at 1E07. (The high nybble is the ink colour in "text" mode only.) Unfortunately, provisions weren't made to change this from BASIC. Ink and paper default are set by the command at 1E45, where the high nybble of the code loaded into A specifies paper, and the low nybble specifies ink colour. I chose to make this 1F instead of F1, so that software designed for the "normal" black-on-white screen would come up looking normal, without first having to change the colour with USR 7881. I changed the border colour to gray (1E07 holds FE) so that it's immediately obvious when you're in TI video mode. (Besides, it looks nice.) Also consider grey as a colour for your usual characters, reserving white for high-lighting.

Peter McM also pointed out that if you're using the Bent ROM as a starting point for burning your EPROM "a", and are using John's tape, you will have to change the command at 0002 so that it again reads LD HL,8000 if you want the normal 16K memory initialization. This is because John's tape POKes a 7F to 0004, assuming the original ROM which loads HL with 7FFF and later increments it. Other than that, Tom's improved ROM code works fine. I especially like the FAST mode initialization, since this keeps your LOAD or SAVE command on the screen while performing tape operations.

Some of John's lower case characters could be improved, particularly "m", "w" and braces. For "m", try the sequence 00 00 68 54 54 54 44 00, and for "w" try 00 00 44 54 54 54 28 00. For the braces, try 00 04 08 08 10 08 08 04 and 00 20 10 10 08 10 10 20. I also modified the symbols to agree with Memotext convention (personal taste). For custom character sets, study John's code at 1E16 (SNGN) to understand how to initialize the VDP's pattern generator in Graphics mode 1. Peter McMullin has designed several custom character sets, let us know if you'd be interested in seeing these.

Compatibility problems with existing software presently seem limited to XFORTH (and also, of course, the TREE-FORTH EPROM), though there may be a very few others. All versions of Hot Z work great, except for a problem when using its built-in tape routines. This is because it jumps into the middle

of John's patch in the end-check routine. The fix is the same for all versions, and simply entails changing JP 0206 to JP 0201 (making this change doesn't affect operation in normal video mode). The address of this command varies according to the version. In the standard Hot Z II, 16K and 64K versions, this is at 5B11. In Hot Z-E V1.72 this is at 328A. If you have a different version, write JP 0206 somewhere in memory and use the string-search command to locate this statement.

Lastly, a mention of some TMS9918A oddities I've discovered. Presumably these apply to other such boards also (e.g. ColoursIn and Kolorworks). Colours (i.e. not white or grey) have a decidedly grainy appearance. If you plot a horizontal line (as with JOBASIC), you'll find that there is a 3:2 pixel relationship. Every third pixel plotted does not actually produce a new pixel on the screen; instead, the last pixel plotted simply gets a little wider. Oddly enough, this is more pronounced on monochrome monitors.

Some systems have initialization trouble, and come up with a blank screen after reset. This appears to be a loading problem, as it goes away when you replace all possible LS chips with HCT types.

White and grey characters have a multi-color "rainbow" look on colour monitors. This is due to the TMS9918 chip itself; TI simply could not include the necessary "trap" (filter) that would be necessary to correct this. Peter McMullin is contemplating working up a "retrofit" board that employs an RGB version of this chip, plus the necessary trap and other circuitry to give composite video output. We'll let you know if/when this materializes.

If you want more information on the software aspects of this amazing chip and truly unique add-on, LET US KNOW! We'd be happy to bring you the "gory details", but only if YOU say you want them!

SALE! 43% OFF!
WOW!

reg.
\$69.95

for ALL T/S
computers

**IN/OUT
PORTS**
SALE! \$39.95

Experiment with computer
sensing and control

The ideal port for sending signals from the outside world into your computer. Or use it to turn external devices on or off under computer control. 8 input lines, 8 outputs, plus 2 handshake strobes. Solderless female connectors. Now available at tremendous savings!

Thomas B. Woods

P.O. Box 64, Jefferson, NH 03583

BASIC TOPICS: Use Logic

John C. Danielson

(NOTE: While the author refers to the TS1000 in his examples, the information put forth in this article applies to any computer using BASIC- even the megabuck mainframes.-ed.)

The topic of logic applied to Sinclair 1000 BASIC is exciting in several ways:

- 1) Using logic, one can branch a multitude of ways from one program line;
- 2) One can cause alternate PRINT results, again from a single line;
- 3) The programmer can cause the program to make alternate assignments of value to variables from a single LET command;
- 4) The operations above will be faster and more memory efficient than a string of simple IF statements.

The logic this article will explore is a blend of two forms; arithmetic and Boolean logic (AND, OR, and NOT) operations. You and I will explore how to apply this blend of both forms to achieve some very powerful results.

Before we begin our dive into logic, consider parentheses ("()"). Parentheses force the TS/1000 to evaluate something between them as if that something had higher processing priority than the surrounding commands or calculations. In other words, the machine evaluates the contents of the inner most parentheses first and then uses the result to attempt to do whatever you told it to outside the parentheses. If you don't understand this, look at the resolutions of expressions later in this article and watch how the parentheses disappear as the computer solves your logic.

How it Works

The easiest way to understand how to use this logic is to look at real examples, so take a minute to look at Figure 1 and then come back to "DISSECTING THE LOGIC".

Figure 1

```

1 REM INITIALIZATION 1-100
200 REM BRANCH MENU ROUTINE
205 LET I=0
210 REM MENU PRINT LINE HERE
220 INPUT I$
230 IF (CODE I$>28 AND CODE I$<
'32) THEN LET I=(VAL I$)
240 GOTO (((I*1000) AND I>0)+(9
000 AND I=0))
1000 REM DISK READ ROUTINE
2000 REM DISK WRITE ROUTINE
3000 REM PRINT FILE ROUTINE
...
9000 PRINT "CANNOT DO THAT, SO T
RY AGAIN."
9010 GOTO 220

```


Dissecting the Logic

Let us presume that line 210 prints a branch menu that looks like this:

MAIN MENU

- 1 LOAD FILE FROM DISK
- 2 WRITE FILE TO DISK
- 3 PRINT FILE

ENTER YOUR CHOICE(1-3)

Just before the menu was printed, line 205 set I to 0. Given this, we need a way to check that what the user enters is a number and that the number is 1 or 2 or 3.

So, line 230 reads: IF (CODE I\$>28 AND CODE I\$<32) THEN LET I=(VAL I\$). This line checks that the input is a number between 0 and 4. (Note that "1" is CODE 29, and "3" is CODE 31.) If the test is false, I remains equal to 0.

The computer resolves the expression in line 230 as follows:

First, it goes to a table in the ROM and pulls out the CODE for the value in I. Let us presume that 1 was entered. The CODE for 1 is 29, so the computer then looks at the expression. 29>28 AND 29<32, so the expression is true. Since this is so, the computer sets I to 1.

The next task is to make the computer branch to the chosen routine or print an error message and give the user another chance. This is accomplished by line 240, which is resolved as follows:

```
GOTO (((I*1000) AND I>0)+(9000 AND I=0))
GOTO ((1000 AND 1)+(9000 AND I=0))
      (true)      (false)
GOTO ((1000)+(9000 AND 0))
      (false)
GOTO (1000+0)
GOTO 1000.
```

The computer goes to the load disk routine.

Let us look at what happens if the user enters an E instead of a number at line 220. The LET in line 230 does not execute, because CODE E is 42, and 42 is not a member of the set (29, 30, 31). I remains equal to 0, thanks to line 205. Line 240 resolves as follows:

```
GOTO (((I*1000) AND I>0)+(9000 AND 1))
      (false) (true)
GOTO ((0 AND 0)+(9000 AND 1))
GOTO (0+9000)
GOTO 9000.
```

The computer prints an error message and gives the user another chance.

Boolean Peculiarities

A few peculiarities of the way the TS/1000 handles AND, OR, and NOT need to be explained here. Let us start with OR.

GOTO (((I*1000) AND I>0) OR (9000 AND I=0)) looks as if it might work like line 240 in Figure 1. However, what the TS/1000 does is (where I=0):

```
GOTO (((I*1000) AND I>0) OR (9000 AND I=0))
GOTO ((0 AND 0) OR (9000 AND 1))
      (false)      (true)
GOTO (0 OR 1)
GOTO 1.
```

The computer looks for a line number 1.

In other words, the OR functions differently than an English "or". If we denote an OR expression as X OR Y, and if X is true, then the machine returns 1 if Y is true or X where Y is false. I have tried to use OR as "or" again and again, until rereading Page 35, Exercise 3(III) in the TS/1000 Manual. In fact, stubborn old me had to review Sharon Aker's "The Logical Operators" in Sync (Volume 3, No. 5) before admitting that the explanation above is really how the TS/1000 handles ORs.

Let's look at NOTs next. Review line 230 in Figure 1, and then look at this rewrite using NOTs:

```
230 GOTO (((I*1000) AND NOT I<=0)+
(9000 AND (NOT I>0 AND I<0)))
```

This statement means the same as the old line 230 did. It is just more complex.

The only use for NOT which I can think of quickly goes as follows:

```
1170 PRINT "ENTER Y TO QUIT"
1180 INPUT I$
1190 GOTO ((2000 AND I$="Y")+(200 (main menu
routine) AND NOT I$="Y"))
2000 STOP.
```

If Y is typed in at the INPUT prompt, the computer treats line 1190 as commanding it to go to line 2000 and STOPS. If anything else is typed in, the computer rePRINTs the main menu.

Mutual exclusivity is required for all this branching to work. OOPS-- translation time. Look at line 1190 in the previous example. Assume a case of idiocy, for the moment. If I\$ could equal "Y" and at the same time not equal "Y", the computer might try to branch to a line 2200. Error B/1190 would result. The past two statements are idiocy because the two conditions are mutually exclusive-- ONLY ONE CAN BE TRUE AT ANY GIVEN INSTANT. Your logical operations should present the computer with mutually exclusive alternatives unless you wish to create chaos.

Since you now have looked at several ANDs, let's glance at them from a different perspective. The rules used to resolve one AND pair are simple. If we denote our pair as X AND Y, and X is true, the following results will occur. The machine will return X where Y is true or 0 where Y is false. If we are looking at X\$ and Y, the machine will return X\$ where Y is true or the null string where Y is false.

Read that last sentence again. One logical extension of that statement is that we can use Boolean logic to control slicing and concatenation of strings. However, this topic needs to be deferred due to its complexity. See you later, with some words concerning strings, error-trapping and structured BASIC.

MORE ABOUT MSCRIPT

Chia-Chi Chao

MSCRIPT uses the true 64 column mode of TS-2068, and therefore, the border does not change color during loading and saving. It would be better if a message is printed after "Start tape & press ENTER" to indicate the acceptance of the keystroke. Also, MSCRIPT does not confirm the Restart option, so hours of work may disappear in a fraction of a second.

Fortunately, MSCRIPT does not use the memory from 32000 to about 36000, and we can put our own routines in that range. The following additions will print saving and loading messages and put a layer of protection for Restart function.

```
8300 CALL A449      ; clear buffer
8303 LD DE,8356     ; address of "Restart" message
8306 LD HL,1700     ; line 17h, column 00h
8309 LD C,1C        ; length=1Ch characters
830B CALL A464      ; print the message
830E CALL A3A8      ; wait for a key
8311 OR 20          ; "lower case lock"
8313 CP 79          ; check if "y" is pressed
8315 JP Z,902B      ; if "y" then jump to restart
8318 JP 9768        ; return to Command level

831B CALL 990E      ; output "Start tape ..."
831E CALL A449      ; clear buffer
8321 LD DE,833F     ; address of "Saving" message
8324 LD HL,1700     ; line 17h, column 00h
8327 LD C,0C        ; length=0Ch characters
8329 CALL A464      ; print the message
832C RET            ; return to Saving routine

832D CALL 990E      ; output "Start tape ..."
8330 CALL A449      ; clear buffer
8333 LD DE,8349     ; address of "Loading" message
8336 LD HL,1700     ; line 17h, column 00h
8339 LD C,0D        ; length=0Dh characters
833B CALL A464      ; print the message
833E RET            ; return to Loading routine

833F - 834A ** SAVING **
8349 - 8355 ** LOADING **
8356 - 8371 Press "Y" to restart Mscript
```

The above routines will be added just after the normal "MSCRIPT" CODE from tape so that both the original program and the modification can be loaded at the same time. Then, these routines will be moved to their appropriate locations before running the program.

```
AEFB LD HL,AF09      ; origin
AEFE LD DE,8300      ; destination
AF01 LD BC,0072      ; length
AF04 LDIR            ; move
AF06 JP 9000          ; jump to MSCRIPT
AF09 - AF7A          ; routines
```

These routines may be poked into the memory with a simple BASIC program, or the mnemonics may be entered with HOT-Z, AROS or version 1.82 (short, high memory). First enter, load or plug in appropriate software, and then load the MSCRIPT code by LOAD ""CODE. Be sure the MSCRIPT has been customized for your printer interface.

```
10 FOR X=44795 TO 44922
20 PRINT X;" ";
30 INPUT A
40 PRINT A
50 POKE X,A
60 NEXT X
```

Decimal Pokes for use with the Basic Loader program shown above

44795	33	44859	154
44796	9	44860	17
44797	175	44861	73
44798	17	44862	131
44799	0	44863	33
44800	131	44864	0
44801	1	44865	23
44802	114	44866	14
44803	0	44867	13
44804	237	44868	205
44805	175	44869	100
44806	195	44870	154
44807	0	44871	201
44808	144	44872	42
44809	205	44873	42
44810	73	44874	32
44811	154	44875	83
44812	17	44876	65
44813	86	44877	86
44814	131	44878	73
44815	33	44879	78
44816	0	44880	71
44817	23	44881	32
44818	14	44882	42
44819	23	44883	42
44820	205	44884	32
44821	100	44885	76
44822	154	44886	79
44823	205	44887	65
44824	154	44888	68
44825	153	44889	73
44826	246	44890	78
44827	32	44891	71
44828	254	44892	32
44829	121	44893	42
44830	202	44894	42
44831	43	44895	80
44832	144	44896	114
44833	195	44897	101
44834	104	44898	115
44835	151	44899	115
44836	205	44900	32
44837	14	44901	34
44838	153	44902	89
44839	205	44903	34
44840	73	44904	32
44841	154	44905	115
44842	17	44906	111
44843	63	44907	32
44844	131	44908	114
44845	33	44909	101
44846	0	44910	115
44847	23	44911	116
44848	14	44912	97
44849	12	44913	114
44850	205	44914	116
44851	100	44915	32
44852	154	44916	77
44853	201	44917	115
44854	205	44918	99
44855	14	44919	114
44856	153	44920	105
44857	205	44921	112
44858	73	44922	116

A few pokes are also needed to route the program to our modifications:

```
POKE 38812,0      ; direct restart routine to
POKE 38813,131    ; routine at 8300h
POKE 39016,45     ; intercept loading routine
POKE 39017,131    ; to routine at 832Dh
POKE 39156,27     ; intercept saving routine
POKE 39157,131    ; to routine at 831Bh
```


The parameters for variables such as window size and page length are initialized when you start or restart the MSCRIPT. To change the initial values, poke the address in the "POKE" column with a desired number.

VARIABLE	POKE	DEFAULT
Window size	36952	64
Page length	36956	60
Page spacing	36960	6
Left margin	36964	0
Line length	36968	60
Line spacing	36972	1

The printer driver supplied for Aerco IF is set for printers which automatically add a line feed after a carriage return, and it's just the opposite for the Tasman IF driver. Although most printers have dip switches for either mode, sometimes it's more convenient to set it by software.

For Aerco IF:

```
POKE 42859,219
POKE 42860,127
POKE 42861,24
POKE 42862,225
```

For both IFs:

```
POKE 42837,0 if printer adds linefeed
POKE 42837,10 if printer doesn't
```

The keyboard decoding method used by MSCRIPT was explained by Mark Fisher in SWN 2/5. However, there are other characters in the ROM which are not accessible in Timex BASIC, but can be printed using

MSCRIPT. Those characters are: CODE 123, CODE 124, CODE 125 and CODE 126. On the Spectrum, these characters are symbol-shifted on F, S, G and A keys, respectively. Please read Fisher's article for more details.

If you have finished the necessary modifications, save the MSCRIPT code a few times by:

```
SAVE "MSCRIPT"CODE 36864,8059
```

Jack Dohaney's FAT MSCRIPT loader (SWN 2/4) greatly enhances the clarity of the characters, but a few changes have to be made. Clear any BASIC program you may have in the computer by typing NEW and ENTER. Type or MERGE in the FAT MSCRIPT program. Change the following lines:

```
5 DATA 017,000,128,213,001, ...
15 ... : SAVE "MSCRIPT"CODE 36864,8059: ...
21 CLEAR 65535: CLS: RANDOMIZE USR 44795
```

The change in line 5 from "253" to "128" will generate the new character set in the "free" memory (32768 - 33535) so that it will not take up our precious text space.

I would also like to note that the HI (Hanging Indent) command must be followed by a number. The "Y" and "N" option is not available on TS-2068. The "/" function in the user defined print code is also not recognized by MSCRIPT. Each print code is treated as an unprintable character, so this can be a problem when you want to use right justification.

HALT RIGHT THERE!

Fred Nachbaur

You ZX/TS users have been told, and told again, "Don't mess with the interrupts" and "Never use the HALT command." The severity of the admonitions seem to imply that a fate-worse-than-crashing awaits those who would tamper with such things.

Many TS2068 users already know that interrupt mode 2 can be quite useful; some of the better Spectrum games use this mode of operation. (See also Tom Woods' "Buggy Software" in a previous issue.) ZX/TSers, take heart; you too can experiment with such exotica, albeit on a more limited scale. (The exception is if you have John Olliger's video upgrade, in which case all interrupt restrictions are removed. Using interrupt mode 2 with this device will be the topic for another article.) The Z80 DI (disable interrupt) and EI (enable interrupt) commands are already in use in most "software hi-res" packages to help manage the hi-res display file.

That leaves the HALT command. When this command is encountered, the Z80 CPU stops running your program. In this way it is like the BASIC STOP command. More precisely, it endlessly executes NOPs (no operation) to keep memory refreshed. So far, it sounds like a pretty useless command, all right. However, your program will continue running after the HALT if the CPU receives an interrupt signal. The interrupt can thus be considered analogous to the BASIC CONT command.

In FAST mode, using the HALT command will most certainly cause the computer to "hang up," unless you supply an external INT signal to get it going again. In SLOW mode, however, the computer's logic chip continuously and automatically generates interrupt signals independent of the CPU. These are what are used to synchronize computer operation with your TV display. Whenever an interrupt occurs, the CPU stops what it's doing and goes off to 0038h to generate the TV picture. As a result, using the HALT command in SLOW mode will stop your program until the next TV frame has been completed.

By now, the usefulness of the HALT command in SLOW mode may be dawning on you. Clever use of this command can help to smooth out moving graphics by synchronizing them with the TV display. As a demonstration, enter a 1 REM line followed by 27 characters, followed by the little decimal loader below:

```
1 REM XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
2 FOR I=16514 TO 16540
3 INPUT A
4 POKE I,A
5 PRINT I,PEEK I
6 NEXT I
```

RUN the program, and INPUT the following values:

```
0, 42, 12, 64, 6, 4, 35, 119, 16, 252, 201, 62, 128,
205, 130, 64, 62, 0, 205, 130, 64, 205, 70, 15, 208,
24, 240.
```

CLASSIFIED

Do you have something to sell? Let everyone know. Just \$2.75 a line will put it here. 32 column (screen) format, include spaces in your text, and send CK. or MO with your listing.

Send to SWN Classified, 602 S. Mill St., Louisville, OH 44641.

NEEDED: One (1) video monitor as donation to church (computer and printer already donated). Contact Floyd Rathbone, 1st Baptist Church of Berwick, 3919 Mount St., Berwick, LA 70342. Phone (504) 384-5241.

JUNKWARE from Thomas B Woods! Over the past 5 years we have collected hundreds of tapes which were returned for one reason or another. Most load. Most have genuine TBW titles. At worst, you get perfectly usable cassettes at a great price. And maybe, you'll get a real bargain! No Guarantees. You buy AS IS. But look at the price!! 6 tapes just \$3 WOW!! The box is overflowing. we've got to get 'em out of here. Add \$1 postage. TBW, Box 64, Jefferson, NH 03583.

Enter RAND USR 16525, and you'll see a rapidly flashing bar at the top of the screen. Notice that, since the routine is not synchronized with the TV display, it looks choppy and "gappy." Stop the routine by pressing BREAK. Now, enter POKE 16514,118. This installs a HALT command at the start of the main printing subroutine. Again enter RAND USR 16525, and press BREAK when you tire of it. Notice how smooth it is now? The black bar (or the bar of spaces that erases it) is not printed until the computer finishes drawing the TV picture, resulting in smooth, consistent operation. POKE 16514,0 to restore it to original.

The time you are allowed depends on the "dead time" between frames, less the overhead required by the NMI handling routine that counts the blank lines. If you're enterprising, you could figure this out by chasing down the NMI routine. But if you're lazy like me, you might prefer to experiment. You can change the number of spaces printed by the demo routine with a POKE 16519,N (N must be less than 33!) The time to print the bar (in T states) until the next HALT is $77 + 26*(N-1)$. The time to print the blanks is 109 T states higher because this is where we check for BREAK. I found that printing 5 spaces works, but printing 6 messes up the synchronization; this means that the maximum on the length of the routine is between 290 and 316 T states.

Thanks to James Hastings-Trew (Callisto Software) for pointing this out.

SyncWare News

CONTENTS

For Your Support....	2
Forum....	3
WORDSINC II.5 & Wordfont	
TS1000 REVIEW....	8
Put Those Extra Banks to Work	
TS2068, Stelling....	9
Basil's Compendium TS1000	
Wentworth....	10
TS2068 Music Maker, Rissel....	13
Fun With Hot-Z II, TS1000	
Nachbaur....	15
TMS9918A Video Upgrade:	
A Follow-up, TS1000	
Oliger, McMullin, Nachbaur... 16	
Use Logic, Danielson....	20
More About Mscript, TS2068	
Chi-Chao....	22
Halt Right There! TS1000	
Nachbaur....	23

FIRST-CLASS MAIL
U.S. POSTAGE
PAID
Jefferson, NH
Permit No. 1

Z.A. Lewis
PO Box 559
Camino, CA

95709

4:4

Editor:
Jeffrey Moore
602 S. Mill Street
Louisville, OH 44641

Technical Director:
Fred Nachbaur
C-12, Mtn. Stn. Broup Box
Nelson, BC V1L 5P1 Canada

Advertising, Subscriptions:
Thomas B. Woods
P.O. Box 64
Jefferson, NH 03583

Submissions, announcements, letters to the editor, and all other material of editorial interest should be sent to the Maryland address.

Copyright 1986 SyncWare News

Subscribe ! \$16.95

For Canada and Mexico please add \$3.00 postage.
All other foreign add \$9.00 Airmail (U.S. funds only)

1 year (6 issues)